

TESI DI LAUREA TITOLO

CARATTERIZZAZIONE DELLA RISPOSTA EMODINAMICA PER LA GENERAZIONE DELLE MAPPE DI ATTIVAZIONE DA IMMAGINI fMRI

Laureando	Lara Candeo
Relatore	Prof. Alessandra Bertoldo
Correlatori	Dott. Emanuela Formaggio Dott. Silvia Francesca Storti

Corso di laurea Specialistica in Bioingegneria

Data Laurea 13 Luglio 2009

Anno Accademico 2008/2009

Autorizzo consultazione e prestito tesi

Indice

Introduzione	4
Capitolo 1 Elementi alla base della risonanza magnetica funzionale	6
1.1 Risonanza magnetica nucleare	6
1.1.1 Sequenze di eccitazione	9
1.2 Risonanza magnetica funzionale, il segnale BOLD	12
1.3 Strumentazione in Risonanza Magnetica	15
Capitolo 2 Data set	20
2.1 Acquisizione dei dati	20
2.2 Preprocessing	21
2.3 Protocollo sperimentale	22
2.4 Data set	23
Capitolo 3 La Risposta Emodinamica	26
3.1 Alcune tipologie di hrf in letteratura	26
3.2 hrf di Handwerker	29
3.2.1 Caratteristiche matematiche	29
3.2.2 hrf base	32
3.2.3 Derivata temporale e derivata dispersiva dell'hrf base	34
3.2.4 Analisi dell'hrf con una variante di $x(t)$	37
Capitolo 4	
Generazione di mappe di attivazione: hrf prefissata e uguale per ogni soggetto	48
4.1 General Linear Model, GLM	48
4.2 Applicazione dei GLM al data set	55
Capitolo 5	
Generazione di mappe di attivazione: hrf specifica per ogni soggetto	58
5.1 Metodi per la generazione di mappe di attivazione con hrf specifica per ogni soggetto	58
5.2 Selezione dei voxel attivi	62
5.2.1 Selezione	62
5.2.2 Raggruppamento	65
5.3 Descrizione del funzionamento della function <i>lsqnonlin</i> di MATLAB®	69
5.3.1 Applicazione di <i>lsqnonlin</i> ai quattro metodi sperimentati	72
5.4 Vincoli e procedure per la determinazione dell'hrf specifica del soggetto	79
5.4.1 Vincoli sulle altezze	79
5.4.2 Vincoli sulle ampiezze	82

Capitolo 6 Risultati	84
6.1 Primo confronto	84
6.2 Gruppo A	91
6.3 Gruppo B	93
6.4 Confronto dei risultati ottenuti con i metodi ottimi	95
6.5 Analisi dei risultati in rapporto alla gravità della lesione nei diversi soggetti	96
6.5.1 <i>Analisi visiva dei risultati</i>	97
Conclusioni	116
Bibliografia e sitografia	118

Introduzione

L'obiettivo di questa tesi è la realizzazione di un software che stimi un particolare tipo di risposta emodinamica caratteristica del soggetto da utilizzare nella generazione di mappe di attivazione. I risultati di tale software sono confrontati con alcune tipologie di risposte emodinamiche fisse non dipendenti dal singolo soggetto presenti in letteratura.

Nel primo capitolo è sembrato rilevante dare un'informazione di carattere generale sui principi fisici alla base della rilevazione dei dati sperimentali e sulla strumentazione necessaria per ottenere tali dati.

Per procedere alla generazione delle mappe di attivazione, scopo finale della tesi, nel secondo capitolo si è ritenuto opportuno trattare le tipologie di protocollo sperimentale, precisando tutti i parametri e le impostazioni iniziali utilizzate per la rilevazione oltre alle prime operazioni di correzione dei dati grezzi. Infine vengono riportate le caratteristiche di ogni paziente del data set.

Il capitolo 3 è dedicato allo studio di una funzione per la risposta emodinamica. In particolare dopo una breve analisi storica delle hrf in letteratura ci si è soffermati sulle nuove caratteristiche dell'hrf implementata nel software.

La parte centrale della tesi è rappresentata dai capitoli 4 e 5. Il capitolo 4 descrive il metodo General Linear Model che sta alla base di tutti i software del presente elaborato. Inoltre si sofferma a trattare su tutti e 4 i metodi con risposta emodinamica standard per tutti i soggetti.

Il quinto capitolo analizza i 4 metodi implementati per la generazione di mappe di attivazione con risposta emodinamica dipendente dai dati del singolo soggetto. Questi metodi ricavano l'hrf specifica attraverso una stima non lineare.

Il capitolo 6 riguarda l'analisi dei risultati sia dal punto di vista matematico, con l'indice di Akaike, sia attraverso uno studio visivo, identificando il metodo che si propone di utilizzare per un eventuale nuovo paziente.

Capitolo1

Elementi alla base della risonanza magnetica funzionale

Nei primi anni '90 è iniziato lo studio di metodi per la generazione di immagini funzionali mediante Risonanza Magnetica (MR), che ha prodotto una nuova metodica denominata functional Magnetic Resonance Imaging (fMRI), la quale oltre a visualizzare la struttura dell'organo analizzato, attraverso i principi della RM, ne valuta la funzionalità mediante un agente di contrasto endogeno, emoglobina deossigenata, o esogeno, ad esempio il gadolinio chelato. A questo proposito tratteremo di seguito i principi alla base della RM e dell'effetto Blood Oxygenation Level Dependent (BOLD), utilizzato per ricavare i data set elaborati nella presente tesi.

1.1 Risonanza magnetica nucleare

A tutti i nuclei atomici si può associare una quantità detta *spin* che descrive le proprietà del moto rotatorio del nucleo stesso, tale quantità è denominata *momento di spin* I assimilabile ad un momento di quantità di moto; inoltre ad ogni particella del nucleo atomico viene associato anche un *numero di spin*, I . Ogni elemento con *numero atomico* Z pari e *numero di massa* A pari ha $I=0$, mentre se A è pari e Z è dispari il numero di spin è un numero intero; i nuclei con A dispari hanno sempre I semintero, cioè $I=1/2, 3/2, 5/2, \dots$, quindi i soli attivi rispetto al fenomeno della risonanza magnetica nucleare sono quelli con $I \neq 0$. L'idrogeno ^1H , che è uno degli elementi più presenti negli organismi viventi, possiede $I=1/2$, cioè due livelli

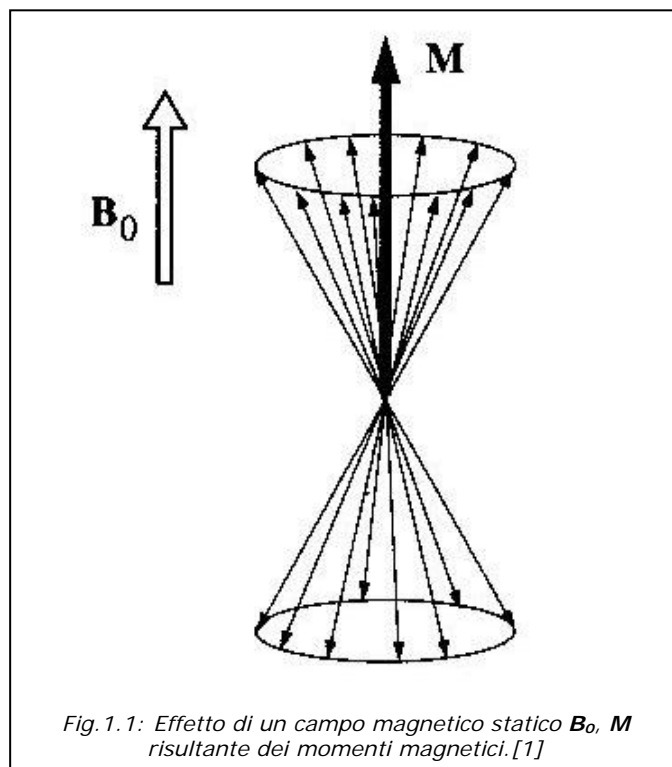
energetici¹, per queste sue caratteristiche è il nucleotide utilizzato per le immagini MR; nel seguito quindi si tratterà il fenomeno riferendosi solo a questo elemento.

Una carica elettrica, un protone, genera *un momento magnetico* μ per effetto del suo moto rotatorio che soddisfa la seguente relazione:

$$\mu = \gamma I \quad (1.1)$$

dove γ è detto rapporto giromagnetico ed è una costante caratteristica di ogni nucleo. Generalmente l'orientazione del vettore μ è casuale ed è dovuta a fenomeni di agitazione termica. Se si sottopone l'atomo di idrogeno ad un campo magnetico statico esterno \mathbf{B}_0 , orientato ad esempio secondo l'asse z, sono possibili due sole orientazioni della componente su z di μ , una concorde a \mathbf{B}_0 , quando il numero quantico magnetico m_I è pari a $1/2$, *spin-up*, l'altra opposta a \mathbf{B}_0 , quando $m_I = -1/2$ *spin-down*.²

Inoltre si ha che su ogni protone agisce un momento meccanico dato dal



¹ I livelli energetici associati al numero di spin I sono pari a $2I+1$.

² La proiezione di μ sull'asse z è data dalla seguente formula: $\mu_z = \gamma m_I \hbar / 2\pi$ dove \hbar è la costante di Planck.

prodotto vettoriale tra $\boldsymbol{\mu}$ e \mathbf{B}_0 da cui risulta per la (1.1) :

$$\frac{d\mathbf{I}}{dt} = \boldsymbol{\mu} \times \mathbf{B}_0 \qquad \frac{d\boldsymbol{\mu}}{dt} = \gamma \boldsymbol{\mu} \times \mathbf{B}_0 \qquad (1.2)$$

La seconda formula descrive il moto di precessione di $\boldsymbol{\mu}$ attorno a \mathbf{B}_0 con una velocità angolare $\boldsymbol{\Omega}$ data dalla *legge di Larmor*:

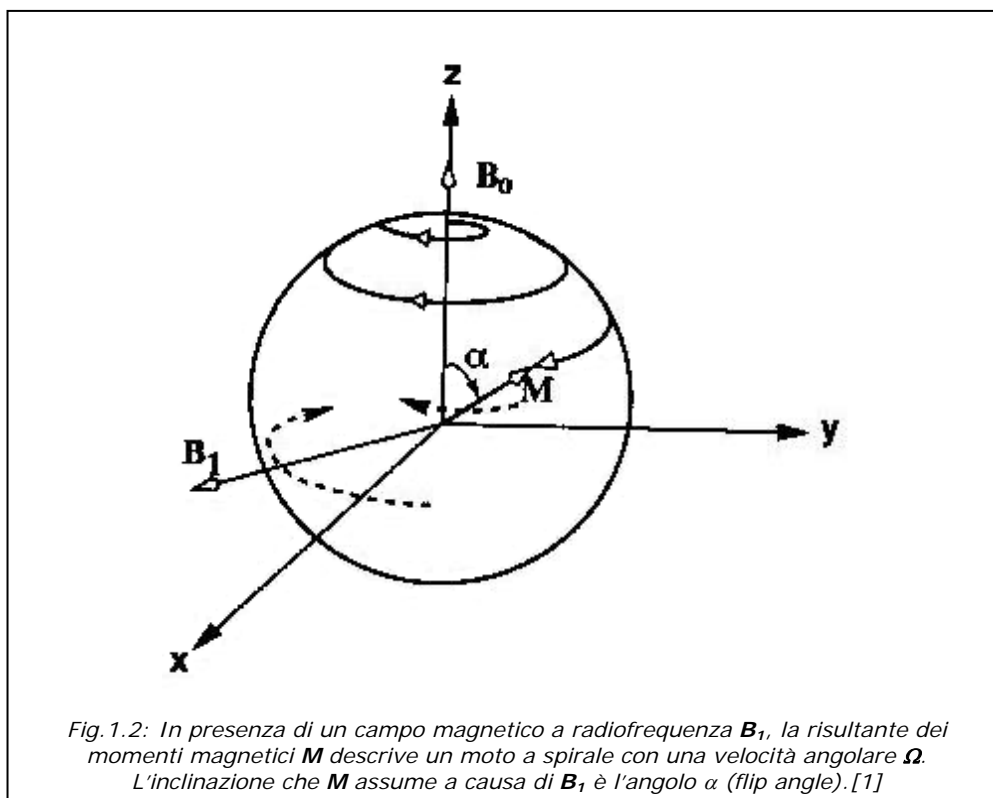
$$\boldsymbol{\Omega} = -\gamma \mathbf{B}_0 \qquad (1.3)$$

Nel campo magnetico statico gli spin dei protoni all'interno dei tessuti tendono ad allinearsi alle linee di forza in modo parallelo, *spin-up*, o antiparallelo, *spin-down*; poiché gli spin allineati in senso parallelo sono in numero superiore, i tessuti vengono a possedere una leggera magnetizzazione totale; chiameremo $\mathbf{M} = \sum \boldsymbol{\mu}_i \neq 0$ la *risultante dei momenti magnetici* dei nuclei atomici. Il campo magnetico \mathbf{B}_0 ed \mathbf{M} sono rappresentate in figura 1.1.

Quindi se si applica ai tessuti un campo magnetico a radiofrequenza \mathbf{B}_1 rotante in un piano ortogonale a \mathbf{B}_0 con pulsazione pari a quella di Larmor, cioè in condizione di risonanza, l'eccitazione a radiofrequenza provoca la rotazione del vettore \mathbf{M} , di un angolo α , attorno a \mathbf{B}_1 dove α è detto *flip angle*³ (figura 1.2). Assumendo una particolare durata di \mathbf{B}_1 si potrà avere α pari a $\pi/2$ o a π , nel primo caso si parlerà di impulso a 90° nel secondo di impulso a 180° a seconda del tipo di immagini che si desidera ottenere.

Terminata l'eccitazione a radiofrequenza il sistema torna spontaneamente all'equilibrio; la componente longitudinale, cioè lungo l'asse z, della magnetizzazione tende a recuperare il suo valore iniziale M_0 , mentre la componente trasversale, sul piano (x,y), decade a zero. Tramite una bobina ricevente viene misurato l'andamento della magnetizzazione nel piano perpendicolare al campo magnetico principale, il segnale misurato è detto *Free Induction Decay* o *FID*. Tale rilassamento avviene con due costanti di tempo distinte. La prima, denominata T_1 , indica la rapidità con cui si ricostruisce la magnetizzazione diretta lungo la direzione del campo principale, e dipende dall'interazione tra protoni e le molecole circostanti, *rilassamento spin-reticolo*. L'altra costante di tempo, detta T_2 , indica la rapidità con

³ Il flip angle è dato da $\alpha = \gamma \mathbf{B}_1 \delta_t$ dove δ_t è il tempo di applicazione di \mathbf{B}_1 .



cui si distrugge la componente di magnetizzazione trasversale in condizioni ideali, e dipende dall'interazione mutua di protoni vicini, *rilassamento spin-spin*. In situazioni reali, la componente trasversa M_{xy} decade a zero con costante di tempo $T_2^* < T_2$ a causa della perdita di coerenza di fase tra i vari protoni del campione osservato. Essendo espressione di proprietà fisiche diverse, queste costanti sono in generale indipendenti l'una dall'altra, e funzioni dell'intensità del campo magnetico B_0 .

In realtà quindi il FID dipende da T_1 , T_2 e da ρ che è la densità protonica del tessuto in esame; è proprio la variazione del peso relativo di queste costanti che caratterizza i diversi segnali rilevati e in particolare il contrasto dell'immagine. Perciò vengono utilizzate delle sequenze di eccitazione opportune che enfatizzano la dipendenza del segnale FID dalle tre grandezze.[1]

1.1.1 Sequenze di eccitazione

Un parametro comune a tutte le sequenze è T_R (Repetition Time), il tempo di ripetizione della sequenza, ossia l'intervallo di tempo fra l'inizio di una sequenza di eccitazione e l'inizio della successiva. Poiché il segnale ottenibile con un singolo FID è estremamente basso, per migliorare il rapporto segnale rumore, si accumulano i risultati ottenuti con più FID successivi. Le sequenze di eccitazione principali di cui

tratteremo sono: *Saturation Recovery* e *Inversion Recovery* che dipendono in particolare da T_1 e da ρ e *Spin Echo* che è dipendente da T_2 e da ρ . Si è omessa la sequenza di eccitazione *Gradient Echo* in quanto riflette maggiormente l'informazione relativa ai grandi vasi, mentre quella del letto capillare che è d'interesse per questa tesi viene rilevata dalla *Spin Echo*.

Saturation Recovery

La sequenza di eccitazione *Saturation Recovery* è quella che enfatizza la dipendenza da T_1 del FID. Tale sequenza consiste nell'applicazione ripetuta di impulsi a 90° distanziati di un tempo di ripetizione T_R , il numero di ripetizioni di tale sequenza deve essere maggiore o uguale a due. Per T_R lunghi rispetto a T_1 il contrasto tra i tessuti dipende dalla sola densità protonica ρ , quindi l'informazione è simile alla tomografia computerizzata (CT); mentre per T_R brevi circa uguali a T_1 il contrasto tra i tessuti dipende essenzialmente da T_1 . L'ampiezza S dei FID successivi al primo è data da:

$$S(t) \propto \rho \left(1 - e^{-\frac{t}{T_1}} \right) \quad (1.4)$$

$$se \ t = T_R \quad S(T_R) \propto \rho \left(1 - e^{-\frac{T_R}{T_1}} \right)$$

Inversion Recovery

La sequenza *Inversion Recovery* è caratterizzata da un impulso iniziale a 180° , che provoca l'inversione della magnetizzazione \mathbf{M} e, dopo un tempo di interpulso T_i , da un impulso a 90° , che consente di rilevare la componente longitudinale parzialmente ripristinata durante T_i . L'ampiezza del FID è data da:

$$S(t) \propto \rho \left(1 - 2e^{-\frac{T_i}{T_1}} + e^{-\frac{T_R}{T_1}} \right) \quad (1.5)$$

$$se \ T_R > 5T_{1MAX} \quad S(t) \propto \rho \left(1 - 2e^{-\frac{T_i}{T_1}} \right)$$

Il variare di T_i , il tempo che intercorre tra i due impulsi della sequenza, consente di modificare il contrasto; ad esempio se si vuole esaltare i parenchimi rispetto ai liquidi si dovrà scegliere $T_i < T_1$. In particolare un aumento del tempo di ripetizione T_R , quindi di T_i , diminuisce il contrasto dell'immagine.[1]

Spin Echo

Sul piano trasverso il progressivo sfasamento degli spin provoca la scomparsa di M_{xy} e lo smorzamento del FID. I due contributi al dephasing sono il rilassamento spin-spin e la disomogeneità spaziale di B_0 . La costante di tempo reale è T_2^* data da:

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T_2'} = \frac{1}{T_2} + \frac{\gamma \Delta B_0}{2}$$

(1.6)

in cui T_2 è il rilassamento spin-spin e ΔB_0 è la massima disomogeneità di \mathbf{B}_0 . Se $T_2^* \ll T_2$ il decadimento è molto più veloce. La sequenza di eccitazione più utilizzata che compensa l'effetto della disomogeneità di \mathbf{B}_0 è la sequenza *Spin Echo*. E' caratterizzata da un primo impulso a 90° , che ruota tutta la magnetizzazione nel piano trasverso rispetto al campo magnetico applicato, e produce un segnale FID che tende a zero con la costante di tempo T_2^* . Successivamente, dopo un tempo $T_E/2$, viene applicato un secondo impulso a 180° , che riallinea gli spin producendo, dopo un ulteriore $T_E/2$, un eco del FID la cui ampiezza S è data da:

$$S(t) \propto \rho \left(1 - 2e^{-\frac{(T_R - T_E)}{T_1}} + e^{-\frac{T_R}{T_1}} \right) e^{-\frac{T_E}{T_2}} \quad (1.7)$$

$$\text{se } \frac{T_E}{2} \ll T_R \quad S(t) \propto \rho \left(1 - e^{-\frac{T_R}{T_1}} \right) e^{-\frac{T_E}{T_2}}$$

dove T_E è detto *tempo d'eco*; i parametri T_E e T_R sono scelti dall'operatore, mentre T_1 , T_2 e ρ sono costanti caratteristiche del tessuto.[1]

Se si sceglie $T_R \gg T_1$ si ottengono immagini che non dipendono da T_1 , come denota tale formula:

$$S(t) \propto \rho \left(1 - e^{-\frac{T_R}{T_1}} \right) e^{-\frac{T_E}{T_2}} \approx \rho e^{-\frac{T_E}{T_2}} \quad (1.8)$$

tali immagini vengono dette T_2 -pesate o T_2 -weighted. Se $T_E \ll T_2$ l'immagine non dipende da T_2 e l'ampiezza del FID sarà pari a:

$$S(t) \propto \rho \left(1 - e^{-\frac{T_R}{T_1}} \right) e^{-\frac{T_E}{T_2}} \approx \rho \left(1 - e^{-\frac{T_R}{T_1}} \right) \quad (1.9)$$

questo tipo di immagini sono le T_1 -pesate o T_1 -weighted; infine se $T_E \ll T_2$ e $T_R \gg T_1$ si hanno immagini che non dipendono da T_1 né da T_2 dette immagini ρ -pesate o immagini di densità protonica, in questo caso l'ampiezza del FID è data da:

$$S(t) \propto \rho \left(1 - e^{-\frac{T_R}{T_1}} \right) e^{-\frac{T_E}{T_2}} \approx \rho \quad (1.10)$$

Si può ricostruire la dipendenza del segnale da T_2 con l'acquisizione di più echi successivi, ciò permette una stima di tale costante che non era possibile con il singolo FID.[1]

1.2 Risonanza magnetica funzionale, il segnale BOLD

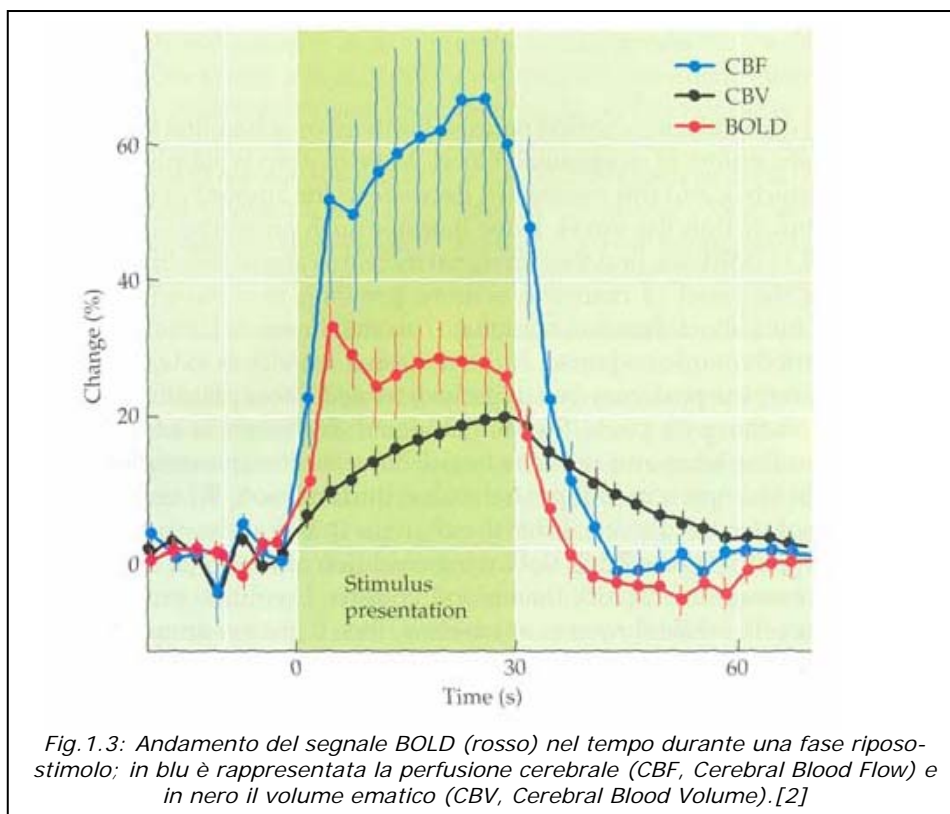
Fino alla fine degli anni '80, il decadimento dei segnali associato a disomogeneità del campo magnetico locale, detto rilassamento T_2^* , era considerato alla stregua di un 'disturbo' e rappresentava una limitazione nelle immagini MR. Per mitigare questo effetto, veniva utilizzata o la tecnica Spin Echo, nella quale una seconda rifocalizzazione degli impulsi RF successiva all'impulso iniziale rimuoveva

lo sfasamento, oppure veniva ridotto il più possibile l'intervallo temporale tra l'impulso RF di eccitazione e l'acquisizione del segnale.[15]

Il riconoscimento che la presenza di sostanze paramagnetiche nel sangue potessero agire come un marker vascolare producendo un contrasto utile, ha portato a nuove applicazioni della tecnica MRI (Magnetic Resonance Imaging). Si è quindi cominciato ad utilizzare, senza sequenze di rifocalizzazione, un intervallo temporale molto più lungo tra l'eccitazione e l'acquisizione del segnale (20-80 ms). Inizialmente l'agente di contrasto paramagnetico, DTPA-Gd, veniva iniettato nel flusso sanguigno attraverso una vena della gamba. Una frazione di millimole per chilogrammo di peso corporeo di questo agente di contrasto è sufficiente a produrre una riduzione del segnale da parte del tessuto che circonda i vasi sanguigni di circa il 40% quando il bolo attraversa il vaso interessato. Si applicò questa tecnica per ottenere il flusso sanguigno cerebrale locale come rapporto tra il volume del sangue ed il tempo medio di transito. Nei primi studi sull'argomento venne esaminato il passaggio del mezzo di contrasto nel cervello di topi e di cani, e furono utilizzati sempre più frequentemente le tecniche di immagini MRI ultraveloce note con il nome di Echo Planar Imaging, EPI, introdotte da Mansfield (1977), le quali permettono di acquisire un'immagine completa in meno di 100 ms e quindi di ottenere 'istantanee' della distribuzione del mezzo di contrasto mentre attraversa il cervello.[15]

In seguito l'MRI è stata applicata a uno studio sull'attivazione funzionale sull'uomo. L'esperimento riguardava la somministrazione di uno stimolo visivo ad un soggetto al quale era stato iniettato un mezzo di contrasto. Le immagini MRI venivano acquisite ogni 0.75 s lungo il piano del solco calcarino in modo da identificare e seguire il passaggio del bolo. Integrando l'intensità dell'immagine relativa a successivi intervalli temporali, sono state ottenute delle stime del volume relativo di sangue e sono state comparate, attraverso un processo di sottrazione dell'immagine, con quelle ottenute quando il soggetto era a riposo. In questo modo è stato osservato un evidente aumento, fino al 30%, di volume sanguigno nella corteccia visiva primaria VI.[15]

Ogawa et al. (1990) e Tumer et al. (1991) dimostrarono in esperimenti condotti su animali da laboratorio che con campi magnetici statici B_0 maggiori di 1.5 T e immagini pesate T_2^* , è possibile misurare un segnale dipendente dal livello di emoglobina deossigenata presente nella regione cerebrale. Questa osservazione derivava dal fatto notato da Faraday e misurato sperimentalmente da Pauling e Coryell (1936) che la deossi-emoglobina, dHb, è paramagnetica ed ha una suscettività magnetica maggiore del 20% rispetto all'emoglobina completamente ossigenata che è diamagnetica. La deossi-emoglobina quindi può essere utilizzata come mezzo di contrasto endogeno. Tale comportamento dell'emoglobina, Hb, è dovuto ai quattro atomi di ferro che ne compongono ogni molecola; nella dHb infatti, il ferro dell'eme ha ancora dei legami liberi, quindi se sottoposto al campo magnetico statico della risonanza magnetica crea una disomogeneità di campo che riduce così i valori di T_2^* con conseguente diminuzione del segnale da esso dipendente. Tale fenomeno è noto come *effetto BOLD*, è alla base delle variazioni del segnale RM dovuto ad un aumento dell'attività neuronale nei singoli voxel del cervello a seguito di uno stimolo sensoriale, motorio o cognitivo. L'attivazione di una zona cerebrale provoca l'aumento del consumo di ossigeno; da ciò consegue un incremento dell'apporto di ossigeno dovuto ad un temporale aumento del flusso sanguigno e del volume dei vasi. Si riscontra un apporto di ossiemoglobina superiore al fabbisogno metabolico



richiesto. L'andamento temporale del segnale BOLD è rappresentato in figura 1.3; si nota che è costituito da una caduta di segnale dovuta ad un iniziale aumento di concentrazione di Hb deossigenata, conseguenza dell'estrazione di ossigeno dai capillari arteriosi, fa seguito un incremento di segnale dovuto all'aumento del flusso locale di sangue ossigenato nella sede attivata, che diluisce l'emoglobina deossigenata facendone diminuire la concentrazione. Con il passare dei secondi il metabolismo cellulare fa estrarre ulteriore ossigeno dai capillari, provocando così un progressivo aumento di concentrazione di deossiemoglobina, con conseguente caduta del segnale, che dopo aver raggiunto un minimo riprende ad aumentare fino a stabilizzarsi sul livello precedente all'attivazione.[15]

Si ricordi infine che il segnale misurato in MRI è generato dai nuclei degli atomi di idrogeno delle molecole di acqua e non dall'emoglobina di per sé che non può essere misurata dalle bobine normalmente utilizzate in MRI.

1.3 Strumentazione in Risonanza Magnetica

L'apparecchiatura per immagini MR (Fig. 1.4) è costituita da:

Magnete principale

Il magnete principale è il componente più grande e costoso dello scanner, e tutto il resto può essere considerato ausiliario ad esso. La sua funzione è quella di creare un campo magnetico \mathbf{B}_0 statico e omogeneo nello spazio e nel tempo. La specifica più importante di un magnete per l'imaging a risonanza magnetica è l'intensità del campo prodotto. Campi magnetici di maggiore intensità aumentano il rapporto segnale rumore (SNR) dell'immagine permettendo risoluzioni più alte o scansioni più rapide. Tuttavia, intensità più alte richiedono magneti più costosi e con costi di manutenzione più elevati, oltre ad avere bisogno di misure di sicurezza più accurate. Al momento campi magnetici a 1.5 T sono considerati un buon compromesso tra costo e prestazioni per l'uso clinico generale tuttavia stanno introducendo apparecchiature con campi magnetici a 3T soprattutto per applicazioni di risonanza magnetica funzionale cerebrale o per l'imaging cardiaco. Alcuni studi sono in corso per ciò che riguarda l'imaging sull'uomo con campi magnetici a 7T ed è

attualmente in costruzione uno scanner *whole-body* a 11.7 T, Progetto Neurospin in Francia. Per esperimenti su cavie e piccoli animali vengono utilizzati campi magnetici fino a 17 T.

Un parametro altrettanto importante per valutare la qualità di un magnete è la sua omogeneità: le fluttuazioni nell'intensità del campo all'interno della regione osservata dovrebbero essere inferiori allo 0.001%.

Si trovano attualmente in commercio tre tipologie di magnete:

Il *Magnete permanente*: è un magnete convenzionale costituito da materiale ferromagnetico, come l'acciaio, può essere utilizzato per ottenere il campo principale. Magneti di questo tipo sono estremamente ingombranti, hanno un peso che può superare le 100 tonnellate, però una volta installati necessitano di pochi costi di manutenzione. I magneti permanenti possono raggiungere solo intensità di campo limitate, normalmente inferiori a 0.4 T, ed hanno stabilità nel tempo ed omogeneità non eccellenti. Pongono inoltre problemi di sicurezza, in quanto il campo magnetico non può essere mai disattivato.

L'*Elettromagnete resistivo*: si tratta di un solenoide di cavo di rame. I vantaggi di questo tipo di magnete sono il basso costo, ma l'intensità di campo è limitata e la stabilità scarsa. L'elettromagnete richiede una corrente elettrica notevole per mantenere attivo il campo, il che lo rende costoso da utilizzare. Questa soluzione è in generale obsoleta.

L'*Elettromagnete a superconduttore*: quando una lega di niobio-titanio è raffreddata da elio (He) liquido a 4 K, essa diventa superconduttiva, cioè riduce la propria resistenza elettrica a zero. Costruendo un elettromagnete con cavo superconduttivo, è possibile ottenere intensità di campo molto alte con ottime caratteristiche di stabilità. La costruzione di un tale magnete è estremamente dispendiosa come pure l'He per il suo raffreddamento che è inoltre molto difficile da maneggiare. Tuttavia, nonostante il costo, magneti a superconduttore raffreddati ad He sono i più comunemente usati negli scanner moderni. Nonostante l'isolamento termico, il calore presente nell'ambiente attorno allo scanner causa una lenta ebollizione ed evaporazione di He liquido. Di conseguenza è necessario un regolare rabboccamento del suo livello. Per questo motivo può essere usato un criostato per

ricondensare l'He che evapora. Sono anche attualmente disponibili scanner privi di raffreddamento ad He, in cui il cavo del magnete è direttamente raffreddato dal criostato.

I magneti principali sono disponibili in diverse forme. I magneti permanenti sono più frequentemente fatti a forma di ferro di cavallo, mentre quelli a superconduttore sono in genere toroidali. Tuttavia a volte, vengono usati magneti a superconduttore anche a ferro di cavallo e magneti permanenti quadrati.

Bobine a radiofrequenza

Servono a generare il campo magnetico rotante \mathbf{B}_1 alla frequenza di Larmor.

Bobine di gradiente

Una componente fondamentale di uno scanner per l'imaging sono le *bobine di gradiente*, avvolgimenti in cui la corrente che vi scorre è modulata a seconda delle direttive della sequenza di eccitazione, e che hanno lo scopo di modificare l'intensità del campo magnetico lungo i tre assi spaziali. La loro caratteristica principale è la generazione di campi magnetici che variano linearmente di intensità lungo una direzione e sono uniformi rispetto alle altre due. Si hanno tre gradienti di intensità G_x , G_y e G_z lungo i tre assi. Il gradiente G_x serve per la codifica in frequenza; G_y viene utilizzato per la codifica in fase e G_z per la codifica lungo l'asse z. Se si attiva ad esempio solo la bobina di gradiente lungo l'asse z si avrà all'interno del magnete un campo uniforme in ogni piano (x,y), mentre nella direzione z varierà secondo la formula $\mathbf{B}_0 + G_z z$, dove \mathbf{B}_0 è l'intensità iniziale del campo magnetico e G_z è l'intensità del gradiente misurata in T/m. Valori tipici per i sistemi di gradienti degli scanner attualmente in commercio vanno da 20 mT/m fino a 100 mT/m.

L'effetto delle bobine di gradiente è quello di modificare la frequenza di risonanza dei nuclei in maniera dipendente dalla posizione spaziale. Questo concetto è alla base della generazione di immagini.

Bobine ausiliarie

Attorno al magnete principale si trovano anche altre bobine oltre alle bobine di gradiente, che hanno la funzione di migliorare le caratteristiche del sistema stesso.

Bobine di shimming

Queste bobine hanno lo scopo di creare campi magnetici tali da annullare le imperfezioni e le non linearità del campo magnetico principale, in modo da renderlo più omogeneo e quindi mantenere il più costante possibile la frequenza di Larmor nella zona di cui si vuole fare l'imaging.

Al posto di bobine controllate elettronicamente, o più spesso in aggiunta ad esse, si utilizza anche uno *shimming passivo*, costituito da elementi in materiale ferromagnetico posti nei dintorni del campo magnetico principale, che ne distorcono le linee di flusso.

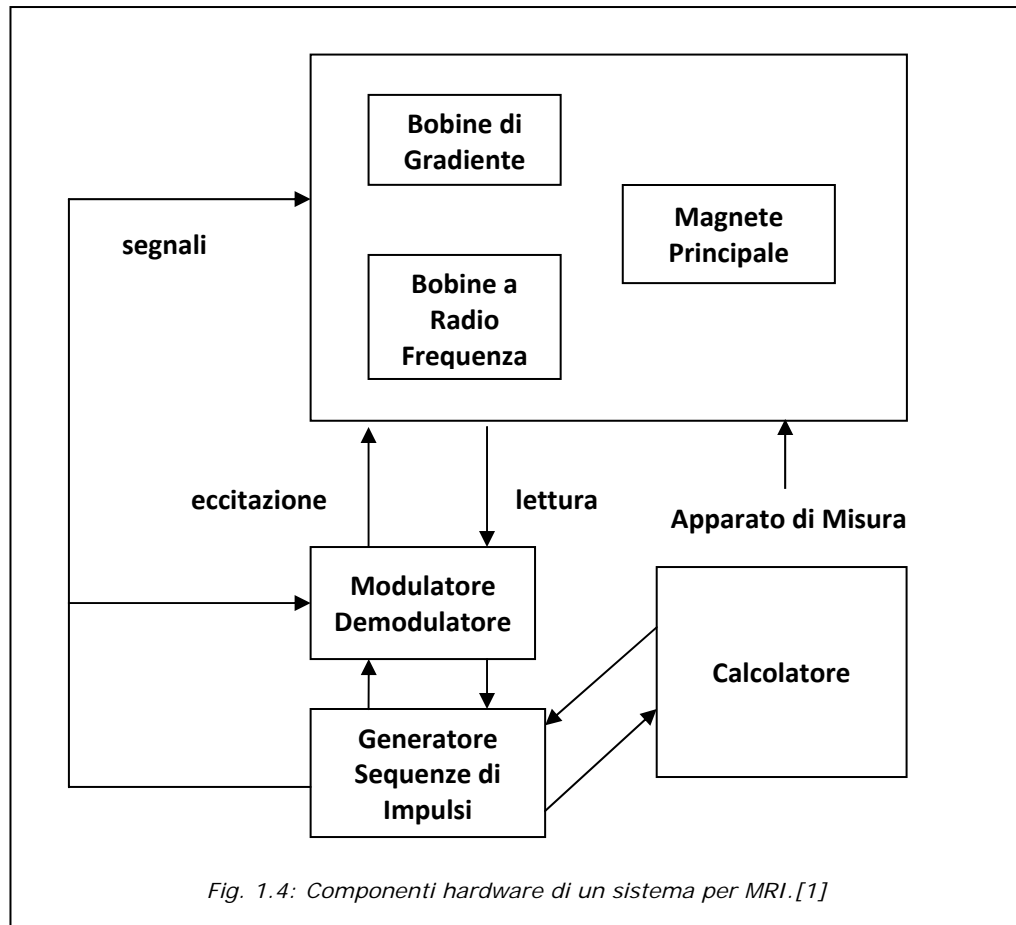
Bobine di schermatura o shielding

Tali bobine possono essere comandate autonomamente, *schermatura attiva*, oppure essere semplici avvolgimenti accoppiati induttivamente con le bobine di gradiente. Lo scopo di questi avvolgimenti consiste nel generare un campo magnetico che si annulli col campo primario o col campo prodotto dalle bobine di gradiente nei punti in cui non è desiderato un effetto magnetico, ad esempio all'esterno del magnete.

Nonostante non siano noti effetti nocivi di un campo magnetico statico sui tessuti organici, almeno alle intensità usate per l'imaging diagnostico, è buona norma cercare di ridurre i campi indesiderati, sia per motivi precauzionali che per la salvaguarda delle apparecchiature elettriche ed elettroniche nei dintorni e all'interno dello scanner stesso.

I campi magnetici variabili creano nei materiali conduttori delle correnti indotte, dette *eddy currents*, letteralmente *correnti a vortice* perché scorrono lungo linee circolari. Esse possono creare interferenza nelle apparecchiature ed effetti biologici negli esseri viventi, in quanto interferiscono con il debole campo elettrico dei neuroni, creando così stimolazioni periferiche dei motoneuroni o, nei casi più

gravi, disturbi temporanei alla vista fino alla stimolazione delle fibre cardiache, con rischio di fibrillazione ventricolare.



Capitolo 2

Data set

Il data set utilizzato nella presente tesi è costituito da 17 soggetti colpiti da ictus in fase acuta e 3 soggetti sani. Le immagini fMRI 3T sono state fornite dal Servizio di Neuroradiologia dell'Ospedale Civile Maggiore di Borgo Trento; le immagini fMRI 1.5T dal Servizio di Radiologia del Policlinico Universitario di Borgo Roma entrambi di Verona.

2.1 Acquisizione dei dati

I 17 pazienti e il soggetto sano numero 20 sono stati sottoposti con lo stimolatore Micromed IRES 600 CH ad una stimolazione elettrica alla frequenza di 3 Hz con durata dell'impulso di 0.1 ms, secondo il protocollo del paragrafo seguente. In particolare con un elettrodo di superficie a ponte è stato stimolato il nervo mediano della mano corrispondente alla lesione nei pazienti, mentre per il soggetto 20 è stato posto sulla mano destra. I dati relativi ai pazienti affetti da ictus sono stati acquisiti attraverso lo scanner MAGNETOM Allegra della Siemens a 3 T.

Una tipica sessione di acquisizione è costituita da una prima fase (Fase I) in cui vengono acquisite le immagini anatomiche T_1 -pesate in sagittale e da una seconda fase (Fase II) in cui si rilevano le immagini fMRI usando la sequenza Echo-Planar Imaging (EPI), in assiale. In tabella 2.1 sono riportati i parametri utilizzati in entrambe le fasi sia per i pazienti che per i soggetti sani.

I dati relativi ai soggetti sani 18 e 19 sono stati acquisiti con lo scanner Symphony della Siemens da 1.5 T. In questo caso i soggetti non sono stati sottoposti

a stimolazione elettrica, ma hanno svolto un compito motorio fasico con la mano destra.

Tab.2.1: Parametri utilizzati nelle sessioni di acquisizione svolte.

Sog.	F a s e	N. volumi	N. fette	Dim. della fetta	Dim. del voxel (mm)	Spazio tra le fette (mm)	Tempo tra le fette (ms)	TE (ms)	TR (s)	Field Of View FOV
1-17,20	I	110	160	256x256	1x1x1	-	-	3	2.3	192x192
	II	110	36	64x64	3x3x3	0.75	72	30	2.6	192x192
18,19	I	100	192	512x512	1x1x1	-	-	3	1.99	256x256
	II	100	36	64x64	3x3x3	-	-	50	3.7	256x256

2.2 Preprocessing

Dopo l'acquisizione, è stato svolto il *preprocessing* con il software Brain Voyager versione 1.2 su tutto il data set a disposizione. Nel nostro caso tale elaborazione si sviluppa nei seguenti quattro punti. Il primo, detto *3D Motion Correction 3DMC*, riguarda la correzione dei movimenti volontari e involontari della testa del soggetto, questi ultimi possono essere causati dalla respirazione o dalla pulsazione cardiaca. Tale correzione è effettuata attraverso la coregistrazione, ossia i volumi cerebrali registrati nei diversi istanti temporali vengono esattamente sovrapposti ad un volume di riferimento dello stesso soggetto. Questa sovrapposizione avviene attraverso tre traslazioni e tre rotazioni, lungo i tre assi (x,y,z). La correzione dà buoni risultati solo per piccoli movimenti ad esempio quelli involontari visti sopra.

Il secondo punto, *Slice Scan Time Correction (SCLAI)*, produce un riallineamento temporale, in quanto ogni fetta è stata rilevata in tempi diversi. Nel nostro caso si sono acquisite le varie fette in ordine crescente e alternato in modo da eliminare ogni possibile correlazione tra le fette adiacenti.

Al fine di aumentare il rapporto segnale rumore SNR nel terzo e quarto punto si sono applicati due filtri. Il primo, *Linear Trend Removal (LTR)*, è un filtro temporale che serve a rimuovere le componenti del rumore periodiche e a bassa frequenza, ad esempio il drift. Il secondo, *Temporal High Pass Filter (THP)*, è utilizzato per rimuovere il rumore fisiologico, nel nostro caso si è mantenuta l'impostazione di default a 3 cicli per time course per togliere l'artefatto provocato dalla respirazione.[19]

Per rendere il processo dell'elaborazione dei dati meno oneroso computazionalmente si vogliono processare solo quei voxel che corrispondono effettivamente al cervello eliminando ad esempio tutti quelli relativi allo scalpo o al bulbo oculare. Perciò si è utilizzata una maschera specifica per ogni fetta di un certo soggetto. Tale maschera è stata ricavata attraverso l'elaborazione delle immagini fMRI in formato analyze con il software Statistical Parametric Mapping versione 5 (SPM5).

2.3 Protocollo sperimentale

In generale i protocolli sperimentali possono essere di due tipi: *Event related* o *Blocked*. Il primo è caratterizzato da due o più condizioni di *task* che si susseguono in modo casuale sia nel tempo, che nell'ordine di esecuzione. Per *task* si intende un compito che il soggetto deve svolgere come ad esempio uno stimolo di tipo visivo, sensitivo, uditivo o un movimento attivo o passivo. Il protocollo che verrà usato anche nella presente tesi è il *Blocked*, il quale è costituito da due o più *task* che si alternano in modo fisso e della stessa durata.

Nel nostro data set vengono utilizzati due tipi di protocolli: il primo per i soggetti dal 1 al 17 e per il soggetto 20 è costituito da 6 stati di *rest*, di 10 volumi ciascuno, alternati a 5 stati di *active* o *event* con lo stesso numero di volumi, per un totale di 110 volumi. Per *rest* si intende la fase di riposo e per *active* o *event* la fase di stimolo; tra un volume e l'altro intercorre il tempo di ripetizione TR pari a 2.6 s (si veda il capitolo 1 per ulteriori dettagli).

L'altro tipo di protocollo è stato applicato ai soggetti 18 e 19 ed è formato da 5 stati di *rest*, sempre di 10 volumi ciascuno, alternati da 5 di *active* per un totale di 100 volumi; il TR è di 3.7 s. Nella pratica il protocollo è rappresentato da un vettore di 0 e 1 dove lo 0 contrassegna i volumi di *rest* e 1 quelli di *active*; la rappresentazione grafica di questo vettore come si può intuire è una successione di onde quadre.

2.4 Data set

La maggior parte dei soggetti del data set sono persone colpite da ictus in fase acuta. L'*ictus* o *stroke* è un evento vascolare acuto caratterizzato da un'assenza o carenza di afflusso di sangue ad una regione dell'encefalo, dovuto ad occlusione, spasmo o rottura di un vaso cerebrale. Nei primi due casi si ha un ictus di tipo ischemico nell'altro di tipo emorragico.

In tabella 2.2 si è riportato per ogni soggetto l'età, il sesso, il lato della lesione, la lesione ossia la classificazione dello *Oxfordshire Community Stroke Project (OCPS)*. La quale classifica il tipo di lesione in base alla gravità e alla posizione dell'occlusione o rottura nei vasi. Nell'ordine dalla meno grave alla più grave si ha:

Lacunar Infarcts LACI o Infarti lacunari sono lesioni con sindromi a minor gravità e con possibilità di recupero, sono determinati dall'occlusione di una singola arteria *perforante profonda*⁴ e possono causare un ictus motorio puro, o sensitivo puro o sensitivo motorio, o un'emiparesi atassica, ossia con mancanza di coordinazione nei movimenti volontari.

Posterior Circulation Infarcts POCI o Infarti del circolo posteriore sono lesioni che presentano sindromi di gravità intermedia con recupero variabile. Sono causate dall'occlusione delle arterie *vertebrali* e *basilari*⁵ con interessamento di lobo

⁴ Arterie perforanti sono le arterie per le strutture mediane, arterie per la corteccia cerebrale, arterie per la sostanza bianca, che nascono dalle arterie portanti.

⁵ Arterie portanti: carotidi interne, cerebrali anteriori, cerebrali medie, vertebrale, basilare, cerebrali posteriori, cerebellari.

occipitale, talamo e cervelletto. Possono causare un deficit motorio e/o sensitivo controlaterale o bilaterale o una disfunzione cerebellare o una emianopsia, ossia una perdita della vista in metà campo visivo.

Partial Anterior Circulation Infarcts PACI o Infarti parziali del circolo anteriore coinvolgono l'arteria cerebrale media nella parte distale dopo la sua diramazione, le zone interessate sono quindi circoscritte e corticali. La sindrome risultante è di media gravità con bassa mortalità e con possibilità di recupero. Si manifesta con deficit sensitivo motorio puro, ma meno esteso di una sindrome lacunare; o con deficit sensitivo motorio accompagnato da emianopsia o compromissione di funzione corticale superiore, altre volte la sindrome si presenta con la sola combinazione di queste ultime due.

Total Anterior Circulation Infarcts TACI o Infarti completi del circolo anteriore interessano l'arteria cerebrale media prossimalmente, prima della sua divisione, con interessamento di strutture cerebrali profonde e superficiali. La sindrome relativa è più grave e caratterizzata da elevata mortalità e disabilità. Manifesta un'emiplegia o un'emianopsia controlaterali alla lesione e un disturbo di una funzione corticale superiore come l'afasia, ossia difficoltà di articolazione o comprensione del linguaggio.[16], [17]

Le altre colonne della tabella riguardano il potenziale motorio evocato (PEM), l'indice National Institute of Health (NIH) e il Barthel Index. Il NIH è un indice che riguarda l'integrità delle varie funzioni cerebrali il cui range va da 0 a 42, dove 0 rappresenta la completa integrità e 42 indica il coma con completa assenza di reazione agli stimoli.

Infine il Barthel index è un indice relativo all'autosufficienza del paziente i valori sono distribuiti su una scala da 0 a 100, dove 100 rappresenta la completa autosufficienza. In generale TACI e POCI hanno indice di Barthel <20; nei TACI e POCI meno gravi e nei PACI e LACI è compreso tra 20 e 75-80; infine nei LACI con danno lieve l'indice supera i 75-80.

Tab.2.2: Data set.

Soggetto	Sesso	Età	Lato lesione	Lesione	PEM	NIH			Barthel Index		
						T0	T1	T2	T0	T1	T2
S1 S.F.	M	55	sx	LACI	presente	1	0	0	100	100	100
S2 Z.N.	F	40	sx	TACI	ridotto	7	4	0	30	90	100
S3 P.S.	M	39	dx	POCI	ridotto	9	0	0	40	100	100
S4 M.L.	F	58	dx	-	presente	7	2	-	40	80	-
S5 B.R.	M	65	sx	LACI	normale	3	1	0	100	100	100
S6 C.A.	M	61	sx	TACI	assente	18	15	10	0	15	40
S7 T.D.	M	57	dx	LACI	-	5	2	-	40	80	-
S8 V.I.	F	85	sx	PACI	ridotto	2	1	1	75	100	100
S9 G.M.	M	58	sx	TACI	assente	14	13	-	0	10	-
S10 M.N.	F	71	sx	TACI	assente	19	19	-	0	25	-
S11 P.M.	F	59	sx	LACI	ridotto	6	3	1	0	45	100
S12 P.A.	F	78	sx	TACI	-	0	0	0	100	100	100
S13 R.E.	M	46	-	-	normale	1	0	0	100	100	100
S14 R.S.	M	69	sx	POCI	assente	12	5	3	5	30	60
S15 G.R.	M	77	sx	PACI	ridotto	13	4	2	30	85	90
S16 Z.B.	F	79	dx	POCI	presente	4	2	0	100	100	100
S17 M.W.	F	65	dx	LACI	-	14	14	-	0	5	-
S18 A.M.	M	34									
S19 M.F.	M	37									
S20 F.E.	F	26									

Capitolo 3

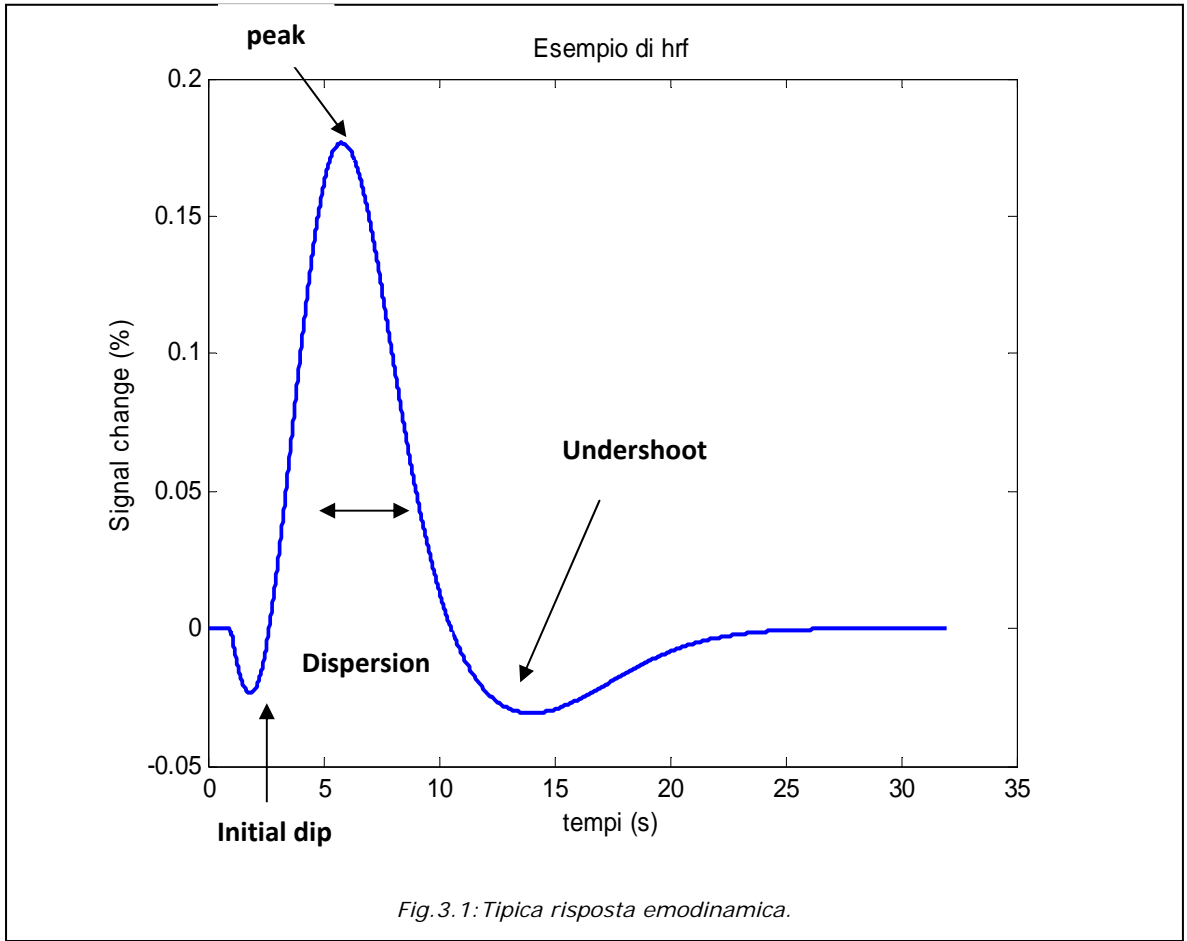
La Risposta Emodinamica

L'*haemodynamic response function* (hrf) è la funzione indice dell'andamento dell'emodinamica cerebrale e come vedremo nel capitolo 4 è la funzione alla base della costruzione del modello che verrà poi confrontato con i dati fMRI ottenuti dai singoli soggetti. Per questi motivi si è scelto nella presente tesi di dedicare una prima parte del capitolo a descrivere le principali varietà di hrf, soffermandoci nella seconda parte ad analizzare in particolare quella che utilizzeremo per l'implementazione dei codici relativi al lavoro svolto.

3.1 Alcune tipologie di hrf in letteratura

La forma della risposta emodinamica si presenta simile nelle regioni sensoriali primarie come pure nelle zone dell'area visiva primaria (V1) e la corteccia somatosensitiva primaria (S1). Tuttavia si è notato che la forma dell'hrf varia particolarmente nelle regioni corticali più alte, probabilmente ciò è dovuto soprattutto alle variazioni del sistema vascolare nelle differenti regioni. Inoltre si è visto che tale risposta è diversa anche da soggetto a soggetto. Per questi motivi si è scelto di ricavare l'hrf da una famiglia di funzioni base.[5]

Nella forma dell'hrf si possono identificare tre fasi: una prima fase è costituita da un *initial dip* cioè una iniziale porzione di curva sotto la linea dello zero, *baseline*, che si forma dopo 1 o 2 s dopo lo stimolo ed è stato rilevato solo in qualche caso con campi magnetici alti. Una seconda detta *peak* dove la curva HRF raggiunge il suo valore positivo massimo dopo 5 o 6 s dallo stimolo e una terza fase l'*undershoot* in cui la curva torna ad essere negativa, raggiunge il suo picco minimo e termina dopo circa 30 s tornando alla baseline (Figura 3.1).[5]



In letteratura sono trattate varie tipologie di hrf, quella più semplice è stata utilizzata nel 1996 da Boynton et al. ed è caratterizzata soltanto dalla fase *peak*, ossia dal solo picco positivo, tale funzione è descritta da:

$$h_B(t) = \begin{cases} \left(\frac{t-o}{d}\right)^{p-1} \left(\frac{e^{-\frac{(t-o)}{d}}}{d(p-1)!}\right)^{p-1} & \text{se } 0 < t < T \\ 0 & \text{altrove} \end{cases} \quad (3.1)$$

dove t è il tempo in secondi, o rappresenta il ritardo iniziale della curva detto *onset*, d è il parametro di scala temporale detto *time-scaling* e p è un parametro intero che rappresenta il ritardo di fase, *phase-delay*. Il ritardo di picco è dato da $p \cdot d$, mentre la dispersione è $p \cdot d^2$. Si noti che la curva $h_B(t)$ ha valore solo all'interno di una specifica finestra di istanti temporali; T rappresenta l'ultimo istante temporale in secondi. Per ottenere un insieme di funzioni $h_B(t)$ è sufficiente incrementare il parametro p . Nel

seguito quando si farà riferimento alla *hrf di Boynton* o *single-gamma* si intenderà una curva costruita come la 3.1 e con parametri $o=0s$, $p \cdot d=6s$, $p \cdot d^2=1$ e $T=32s$. [5]

Nel 1998 Friston et al. presentarono una risposta emodinamica costituita dalla sottrazione di due funzioni gamma:

$$h_c(t) = \begin{cases} \left(\frac{t}{d_1}\right)^{a_1} e^{-\frac{(t-d_1)}{b_1}} - c \left(\frac{t}{d_2}\right)^{a_2} e^{-\frac{(t-d_2)}{b_2}} & \text{se } 0 < t < T \\ 0 & \text{altrove} \end{cases} \quad (3.2)$$

Tali funzioni servono: la prima a modellare la fase *peak* che si avrà in un tempo pari a $d_1=a_1 \cdot b_1$ in secondi; mentre la seconda costituisce l'*undershoot* il cui tempo in secondi verrà rappresentato da $d_2=a_2 \cdot b_2$ e t rappresenta il vettore dei tempi in secondi; questo tipo di *hrf* è denominata *canonical hrf* o *hrf two-gamma*. La parametrizzazione che noi useremo per questo genere di funzione sarà $a_1=6$, $a_2=12$, $b_1=b_2=1$ e $c=1/6$, in quanto consideriamo c come il rapporto tra l'ampiezza dell'*undershoot* e l'ampiezza del *peak*. [2], [6]

Sempre nel 1998 Friston et al. [6], per generare una più vasta gamma di variazioni attorno alla *hrf canonical*, decisero di generare un'*hrf* traslata di un certo tempo δ . Questa traslazione è data dalla combinazione lineare di due funzioni base $h_0(t)$, ossia l'*hrf two-gamma* e $h_1(t)$ la sua derivata:

$$h_1(t) = \left. \frac{\partial h(t, \delta)}{\partial \delta} \right|_{\delta=0} = -\dot{h}_0(t) \quad (3.3)$$

dove $h(t, \delta) = h_0(t - \delta)$ e la notazione con il punto rappresenta la derivata rispetto al tempo. Se δ è piccolo si può approssimare $h(t, \delta)$ con lo sviluppo in serie di Taylor arrestato al primo termine attorno a $\delta=0$:

$$h(t, \delta) \approx h_0(t) + h_1(t)\delta \quad (3.4)$$

Nel 2003 Handwerker et al. partendo dalla *canonical hrf* del software SPM (Friston et al. 1999) riformularono l'*hrf* come riportato nel paragrafo seguente.

3.2 hrf di Handwerker

In questo lavoro di tesi si è scelto di utilizzare l'hrf di Handwerker in quanto, pur avendo una formulazione matematica leggermente più complessa, introduce la possibilità di traslare entrambe le curve positiva e negativa e presenta una maggior flessibilità riuscendo ad adattarsi alle diverse esigenze dei soggetti.

3.2.1 Caratteristiche matematiche

La formulazione matematica che Handwerker et al. [7] proposero nel 2003 è la seguente:

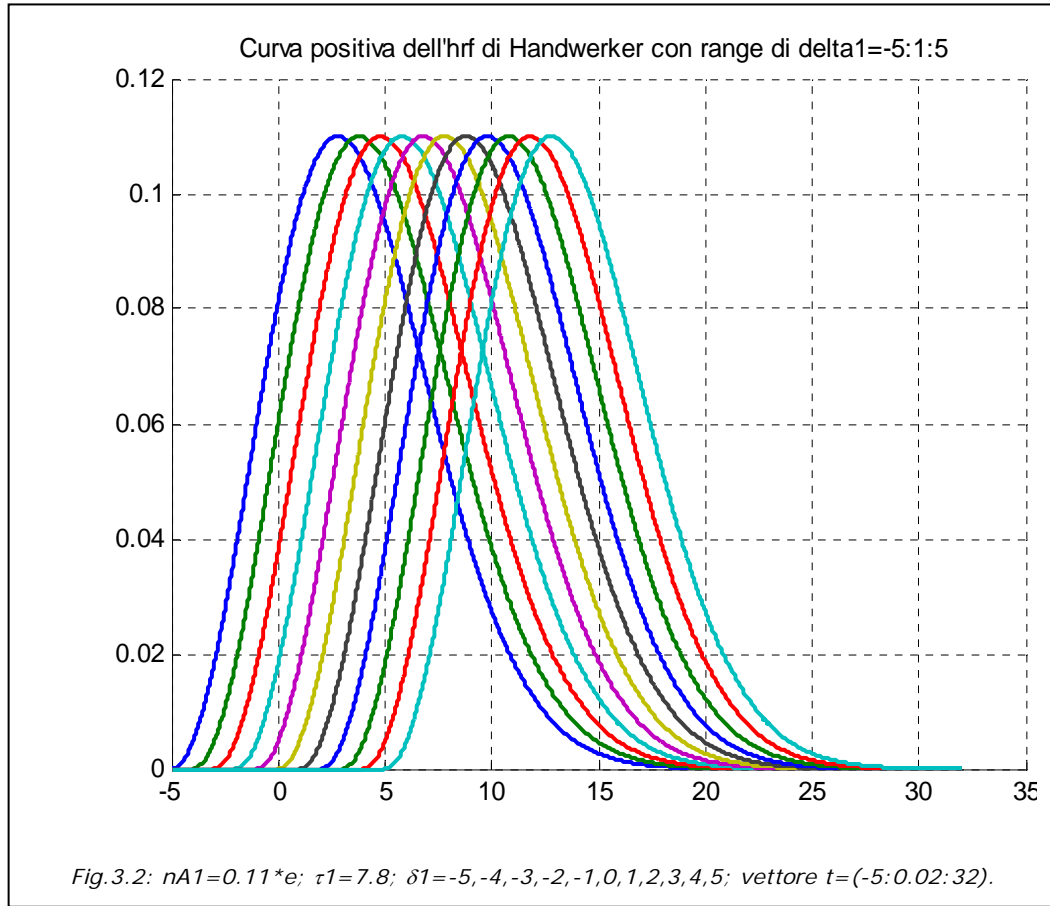
$$y(t) = \underbrace{A_1 * \left(\frac{x(t) - \delta_1}{\tau_1}\right)^2 * \frac{e^{-\left(\frac{x(t) - \delta_1}{\tau_1}\right)^2}}{\tau_1}}_{\text{Curva positiva}} + \underbrace{A_2 * \left(\frac{x(t) - \delta_2}{\tau_2}\right)^2 * \frac{e^{-\left(\frac{x(t) - \delta_2}{\tau_2}\right)^2}}{\tau_2}}_{\text{Curva negativa}} + C \quad (3.5)$$

Nella nostra trattazione la costante C è stata posta a zero in quanto ha la funzione di aggiustare la funzione emodinamica $y(t)$ rispetto alla *baseline* non influenzando sulla sua forma. In tal modo la formula risulta costituita da due componenti con lo stesso comportamento: una parte positiva $y_1(t)$ ed una negativa $y_2(t)$. In particolare considereremo $y(t)=y_1(t)+y_2(t)$ dove $y_1(t)$ e $y_2(t)$ sono prese ciascuna nel loro intervallo di interesse come segue:

$$y_1(t) = \begin{cases} A_1 * \left(\frac{x(t) - \delta_1}{\tau_1}\right)^2 * \frac{e^{-\left(\frac{x(t) - \delta_1}{\tau_1}\right)^2}}{\tau_1} & \text{se } \delta_1 < t < T \\ 0 & \text{altrove} \end{cases}$$

$$y_2(t) = \begin{cases} A_2 * \left(\frac{x(t) - \delta_2}{\tau_2}\right)^2 * \frac{e^{-\left(\frac{x(t) - \delta_2}{\tau_2}\right)^2}}{\tau_2} & \text{se } \delta_2 < t < T \\ 0 & \text{altrove} \end{cases} \quad (3.6)$$

Analizziamo ora gli elementi che compaiono in $y(t)$: il vettore $x(t)$ rappresenta una generica funzione, per ora lo considereremo pari a t , vettore dei tempi espresso in



secondi; i parametri δ_1 e δ_2 rappresentano il ritardo in secondi rispettivamente della curva $y_1(t)$ e di $y_2(t)$. Dall'osservazione delle curve si è notato che τ_1 e τ_2 sono proporzionali all'ampiezza a metà altezza $FWHM^6$ rispettivamente della curva positiva e negativa secondo il rapporto di proporzionalità di 15/13, ossia $FWHM = \frac{15}{13} * \tau$ dove τ è o τ_1 o τ_2 . A_1 e A_2 non sono le altezze dei picchi di $y_1(t)$ e $y_2(t)$. Per stimare l'altezza del picco positivo di $y_1(t)$ si sono calcolati i valori di t per cui la derivata $\frac{dy_1(t)}{dt} = 0$, il valore di t che risulta da questa operazione è $t_{max} = \delta_1 + \tau_1$ ne consegue che il valore del picco positivo è:

$$y_1(t_{max}) = a_1 = \frac{A_1}{\tau_1 e} \quad (3.7)$$

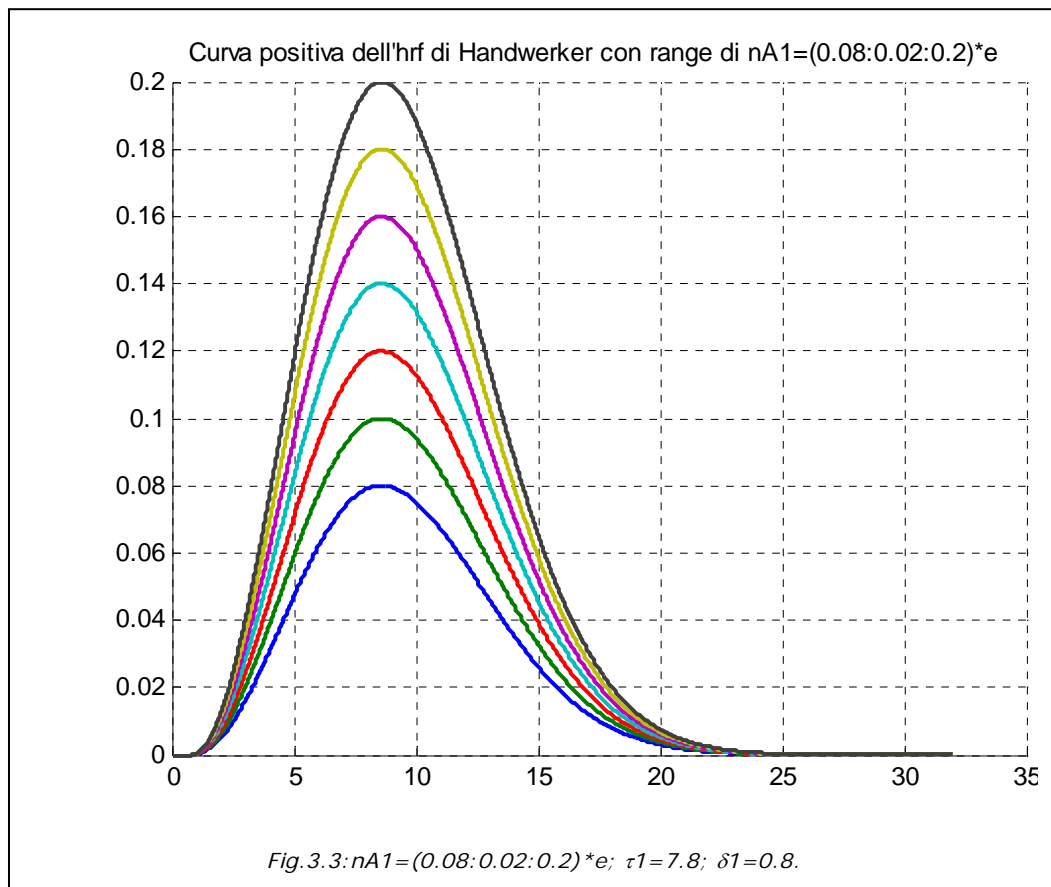
⁶ FWHM sta per Full Width at Half Maximum

si procede in modo analogo per ricavare il valore di picco della curva negativa che

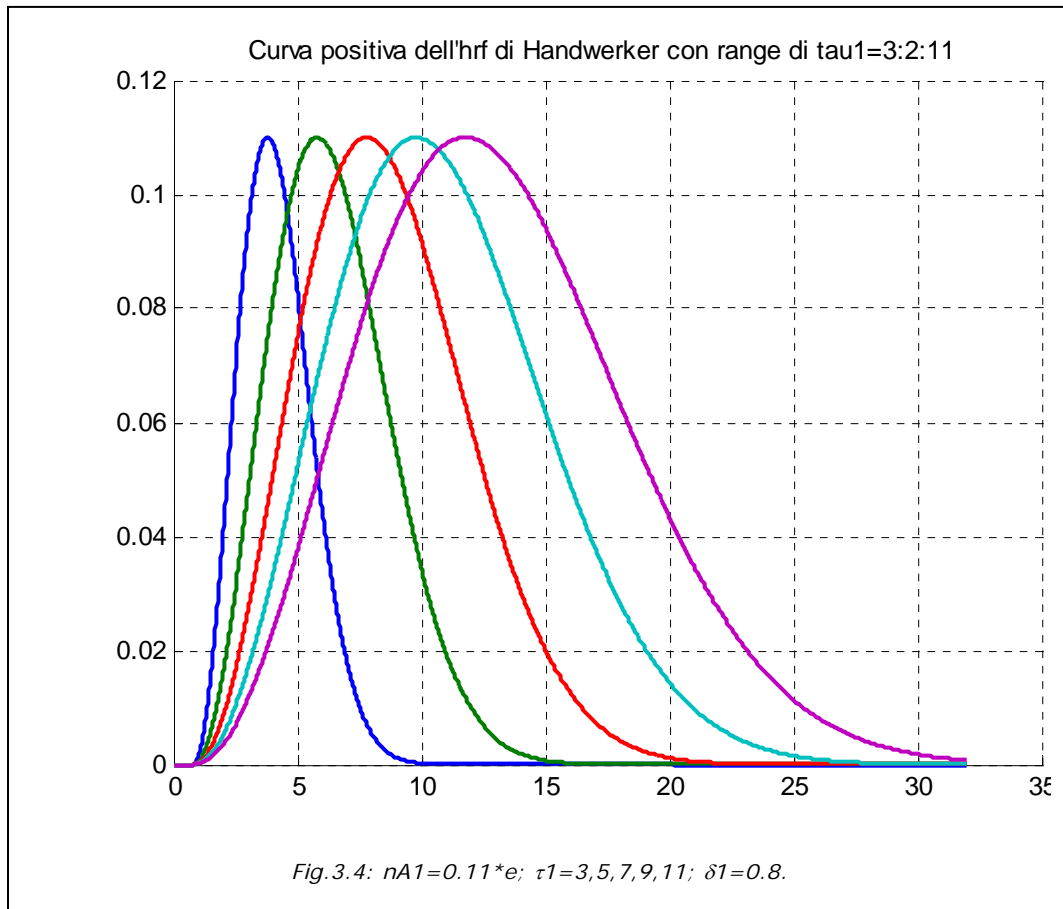
$$\text{sarà: } y_2(t_{min}) = a_2 = \frac{A_2}{\tau_2 e} \quad \text{con} \quad t_{min} = \delta_2 + \tau_2. \quad (3.8)$$

Quindi risulta che $A_1 = nA_1 * \tau_1$ dove $nA_1 = a_1 * e$ e $A_2 = nA_2 * \tau_2$ dove $nA_2 = a_2 * e$.

Poiché le due parti che costituiscono la formula 3.5 hanno andamento simmetrico, possiamo visualizzare il comportamento della sola curva positiva $y_1(t)$, facendo variare singolarmente i parametri a_1 , δ_1 e τ_1 e considerando $t=(0:0.02:32)$,



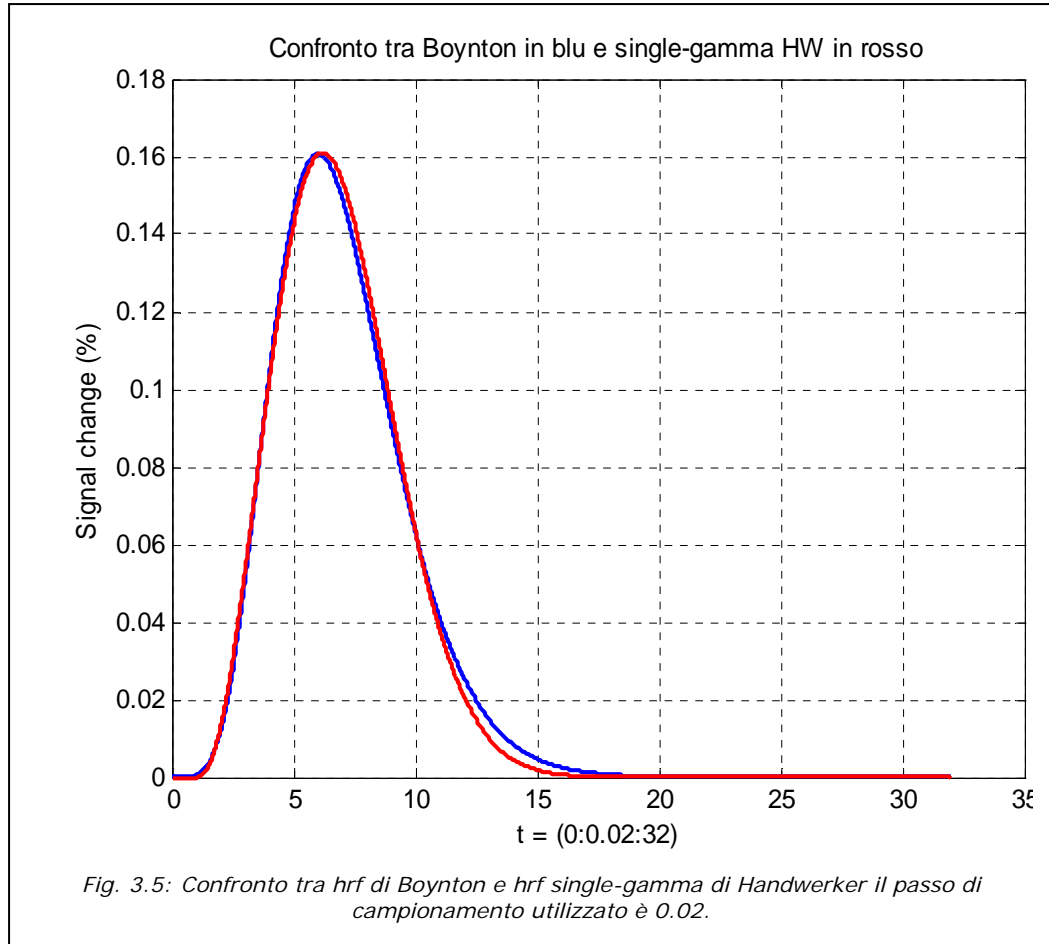
ossia il vettore da 0 a 32 con passo 0.02. In figura 3.2 si vede che al variare di δ_1 tra -5 e 5 la curva $y_1(t)$ trasla della stessa quantità δ_1 ; se δ_1 è positivo lo spostamento avviene a destra dello zero, se δ_1 è negativo la traslazione è verso sinistra. Se si fa variare il parametro a_1 , con δ_1 e τ_1 costanti, l'andamento temporale della curva $y_1(t)$ è quello riportato in figura 3.3; si nota che l'altezza della curva varia della stessa quantità a_1 cioè tra (0.08:0.02:0.2); mentre la posizione e l'ampiezza a metà altezza rimangono costanti. Qualora si intenda variare il parametro τ_1 mantenendo costanti gli altri si avrà una variazione in ampiezza in particolare nella FWHM come riportato in figura 3.4.



3.2.2 hrf base

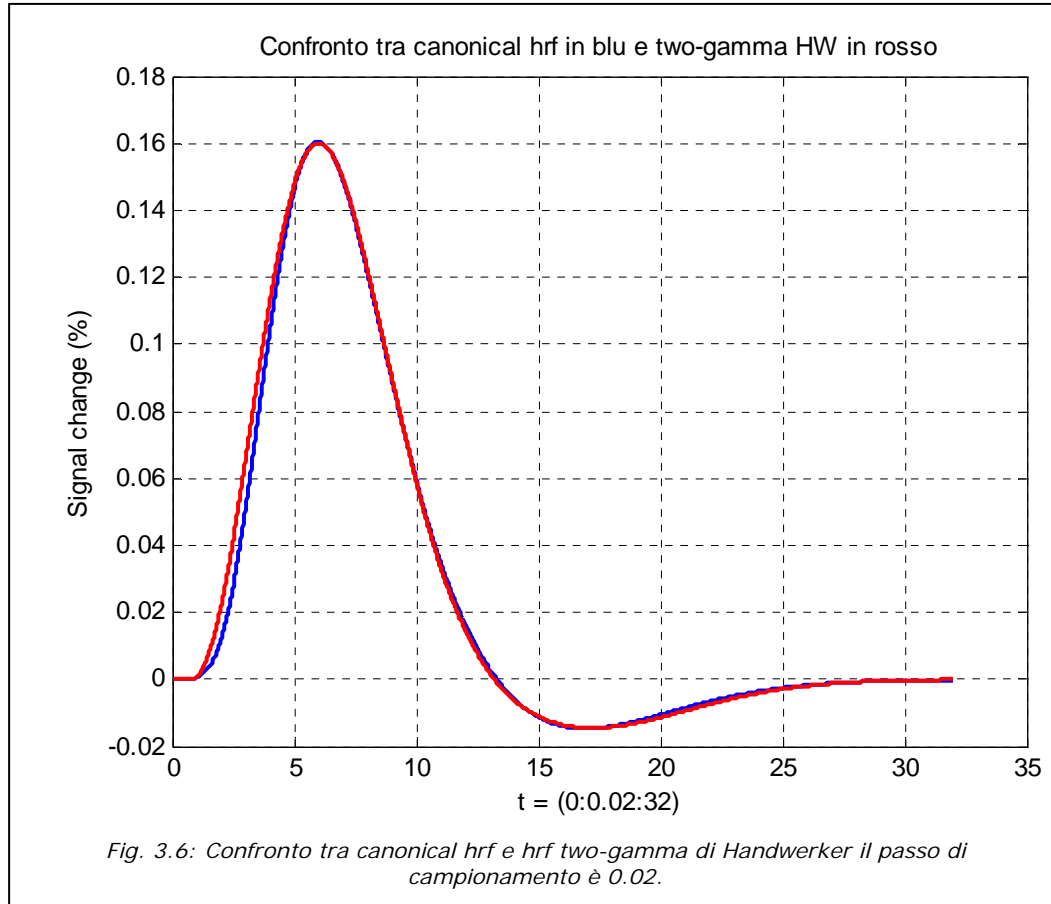
Single-gamma di Handwerker

Se consideriamo l'hrf positiva $y_1(t)$ appena trattata si nota che è del tutto simile all'hrf single-gamma o Boynton. In particolare l'hrf single-gamma di Handwerker, *single-gamma HW*, è stata costruita con i seguenti parametri $\delta_1=1.1$, $\tau_1=5.1$, $a_1=0.161$ applicati a $y_1(t)$ della 3.6 (Figura 3.5).



Two-gamma di Handwerker

La two-gamma di Handwerker che verrà poi indicata come *two-gamma HW* è la funzione $y(t)=y_1(t)+y_2(t)$. Si noti che t_{\max} e $y(t_{\max})$ non vengono alterati in $y(t)$ per effetto della somma se $\delta_2 > \delta_1 + \tau_1 = t_{\max}$ in quanto non vi è sovrapposizione delle due curve per $0 < t < \delta_2$. Per t_{\min} e $y(t_{\min})$ si può riscontrare una leggera alterazione dei due valori in quanto la curva in $y_1(t_{\min})$ è prossima allo zero, ma non è zero; tale alterazione dipende anche da δ_2 e da τ_2 . Il confronto tra la canonical hrf e la two-gamma di Handwerker è riportato in figura 3.6. I parametri utilizzati per la two-gamma HW sono $\delta_1=0.8$, $\tau_1=5.2$, $a_1=0.16$, $\delta_2=9$, $\tau_2=8$, $a_2=0.0147$.



3.2.3 Derivata temporale e derivata dispersiva dell'hrf base

Per dare una maggior versatilità al modello, come già proposto da Friston nel 1998 (vedi paragrafo 3.1), si è calcolata l'hrf utilizzando lo sviluppo in serie di Taylor arrestato al primo termine:

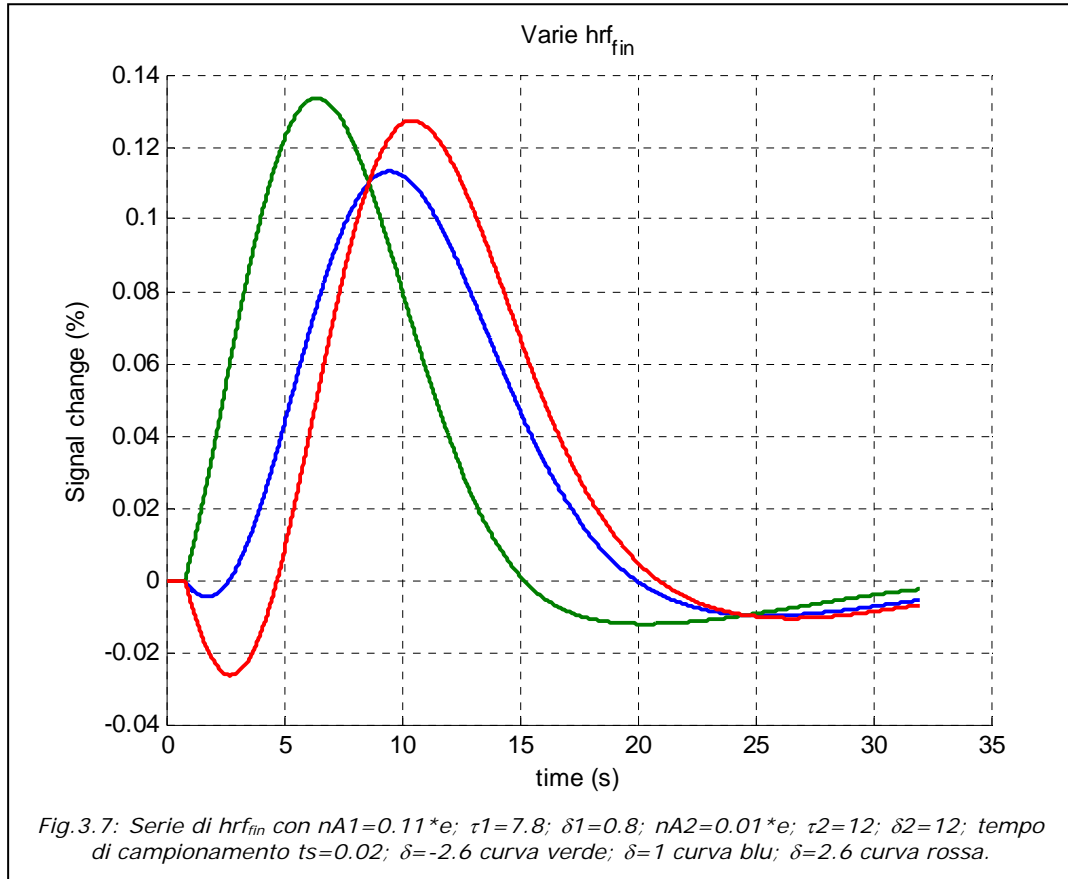
$$hrf_{fin} = y(t) - \frac{dy(t)}{dt} * \delta$$

(3.9)

dove $y(t)$ è la two-gamma HW e la sua derivata rispetto al tempo è:

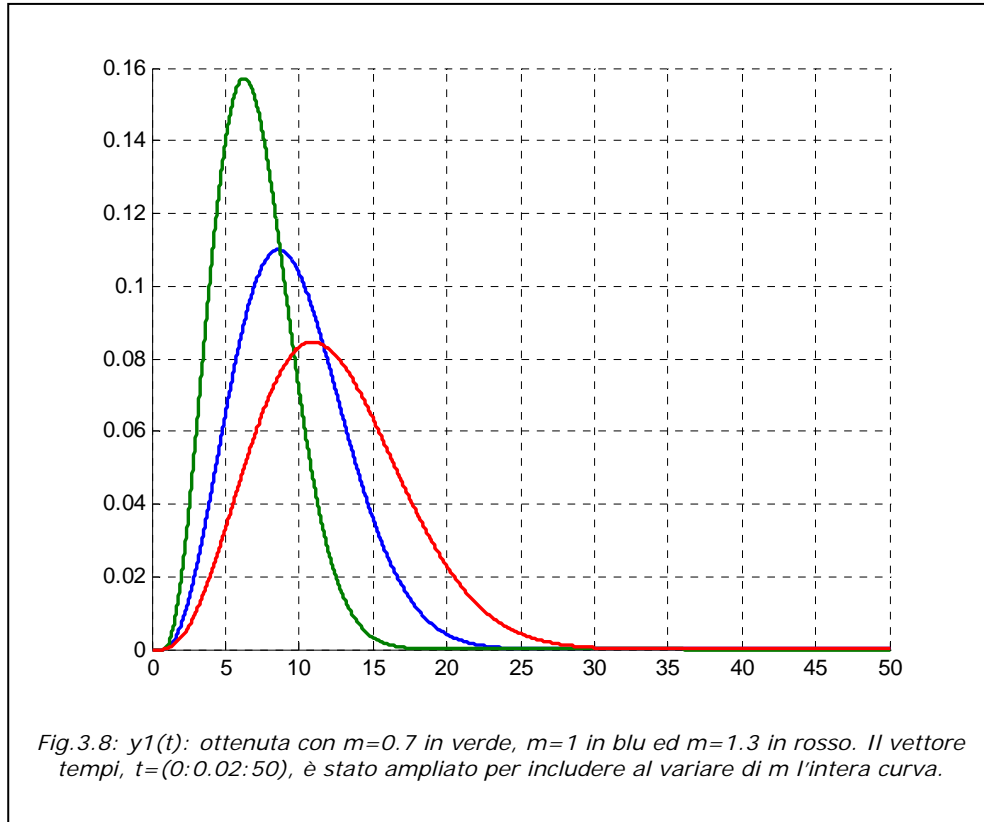
$$\begin{aligned}
\frac{dy(t)}{dt} = & 2 * \frac{A_1}{\tau_1^3} * (x(t) - \delta_1) * \frac{dx(t)}{dt} * e^{-\left(\frac{x(t)-\delta_1}{\tau_1}\right)^2} \\
& * \left(1 - \left(\frac{x(t) - \delta_1}{\tau_1}\right)^2\right) + 2 * \frac{A_2}{\tau_2^3} * (x(t) - \delta_2) * \frac{dx(t)}{dt} \\
& * e^{-\left(\frac{x(t)-\delta_2}{\tau_2}\right)^2} * \left(1 - \left(\frac{x(t) - \delta_2}{\tau_2}\right)^2\right)
\end{aligned}
\tag{3.10}$$

dove, poiché si è posto $x(t)=t$, $\frac{dx(t)}{dt} = 1$.



Con l'introduzione del termine derivativo vi è la comparsa del *dip* iniziale che cambia in ampiezza e altezza in modo dipendente da δ . Un altro effetto della moltiplicazione della derivata con il δ è il ritardo o l'anticipo dell' hrf_{fin} rispetto a $y(t)$, oltre ad una dilatazione in ampiezza (Figura 3.7).

A questo punto analizziamo la dispersione: la dispersione nella canonical hrf comportava una dilatazione o compressione della curva rispetto all'asse dei tempi e una conseguente diminuzione o aumento dei picchi, per mantenere costante l'area.



Nel nostro caso per ottenere lo stesso risultato se si vuole espandere la curva si devono aumentare i parametri che regolano la larghezza dell'hrf, τ_1 e τ_2 , moltiplicandoli per una costante di proporzionalità $m>1$ e diminuire nA_1 e nA_2 , dividendoli per lo stesso m ; analogamente se si vuole comprimere la curva basterà scegliere $m<1$. Così si mantiene costante l'area sotto la curva (si veda la figura 3.8).

Si è sviluppata inoltre anche una derivata *dispersiva*, ossia rispetto ai parametri τ_1 e τ_2 ; la derivata rispetto a τ_1 di $y_1(t)$ risulta:

$$\frac{dy_1}{d\tau_1} = A_1 * (x(t) - \delta_1)^2 * e^{-\left(\frac{x(t)-\delta_1}{\tau_1}\right)^2} * \left[\frac{2 * (x(t) - \delta_1)^2 - 3 * \tau_1^2}{\tau_1^6} \right] \quad (3.11)$$

analoga è la formula della derivata di $y_2(t)$ rispetto a τ_2 . Vedremo nel capitolo 4 come verrà utilizzata la derivata *dispersiva* data da $\frac{dy_1}{d\tau_1} + \frac{dy_2}{d\tau_2}$.

A questo punto è rilevante considerare che essendo la risposta emodinamica legata alle variazioni del *cerebral blood flow* (CBF) e del *cerebral blood volume* (CBV), può essere necessario introdurre una dissimmetria più marcata tra la parte di

curva crescente prima del picco positivo e quella decrescente che segue tale picco. Questa variazione si può ottenere esplicitando $x(t)$ della 3.5 con funzioni opportune.

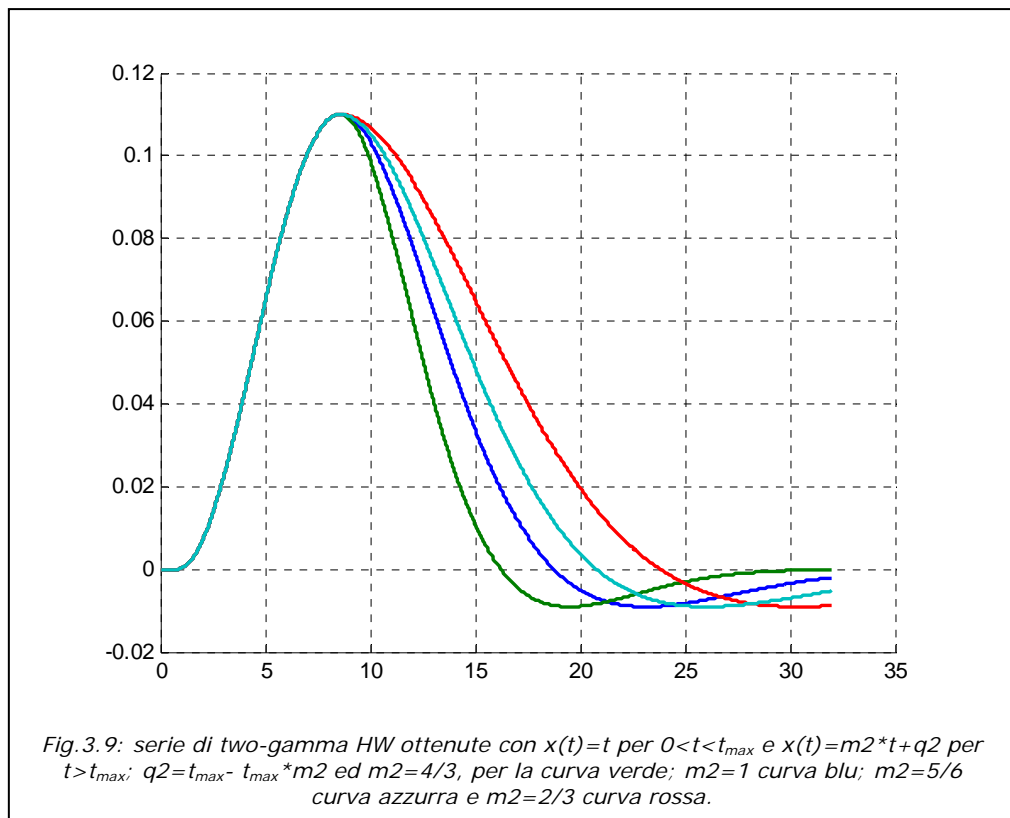
3.2.4 Analisi dell'hrf con una variante di $x(t)$

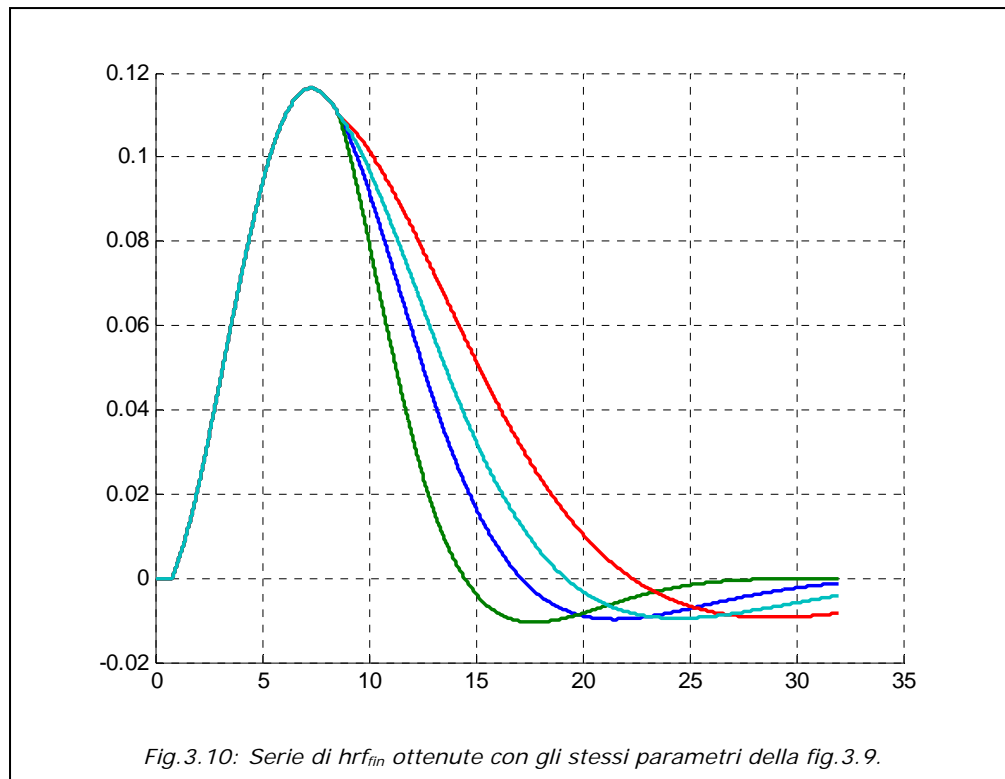
La funzione $x(t)$, già introdotta da Handwerker nell'hrf base (eq. 3.5), che fino ad ora è stata considerata uguale a t , ci dà la possibilità di modificare la simmetria della curva e la dispersione. Se ad esempio sostituiamo in $y_1(t)$:

$$x(t)=m_1*t \text{ con } t \leq t_{\max} \quad \text{e} \quad x(t)=m_2*t+q_2 \text{ con } t > t_{\max} \quad (3.12)$$

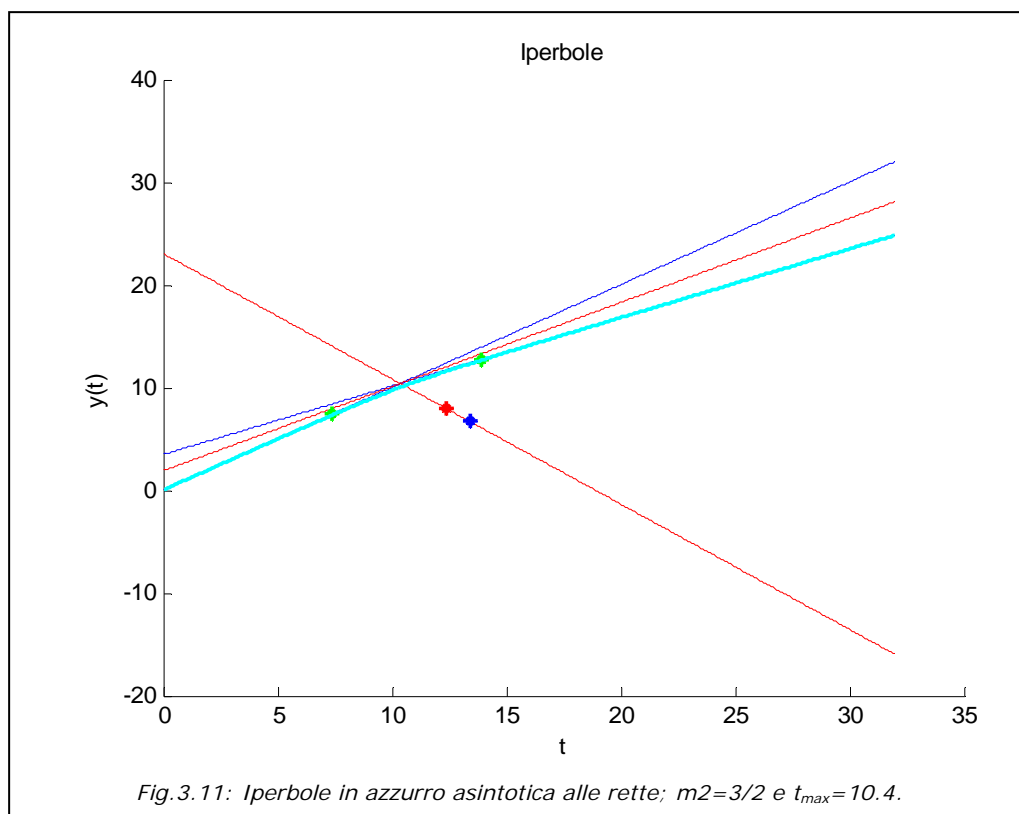
otteniamo una curva come quella rappresentata in figura 3.9. In figura 3.10, dove sono rappresentate le hrf_{fin} , si può notare una discontinuità introdotta dalla derivata temporale nel passaggio repentino tra la funzione $x(t)=t$ e $x(t)=m_2*t+q_2$ con $m_2 \neq 1$; per eliminare tale discontinuità (figura 3.12) una possibile soluzione è la sostituzione della spezzata con un'iperbole asintotica alle due rette (figura 3.11).

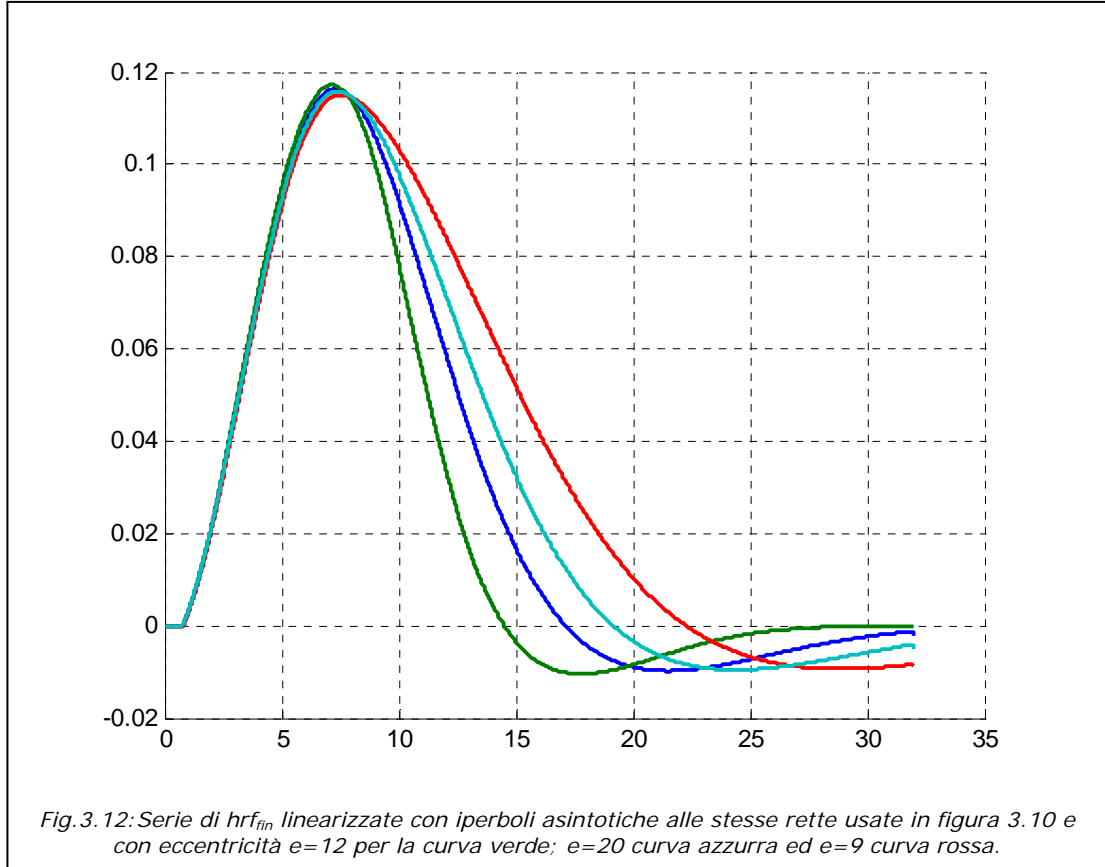
L'iperbole per definizione è il luogo dei punti P tali che $\frac{\overline{PH}}{\overline{PF}} = e$, dove \overline{PH} e \overline{PF} sono rispettivamente le distanze di P dalla direttrice e dal fuoco. L'eccentricità e è costante per tutti i punti dell'iperbole.





Le rette a cui l'iperbole dovrà tendere asintoticamente sono $y_1=m_1*x$ e $y_2=m_2*x+q_2$, le equazioni sono rispetto al sistema di riferimento globale, dove $m_1=1$, m_2 viene passato dall'operatore come parametro mentre q_2 è calcolato in modo che le due rette si intersechino nel punto O_1 di coordinate $(t_{max}, y_1(t_{max}))$. Preso





un punto A sull'asintoto y_1 si è cercato sull'altro asintoto il punto B che soddisfaceva l'equazione $\overline{O1A} = \overline{O1B}$, in modo che la distanza di A da O1 sia uguale alla distanza di B da O1. Si calcola quindi un punto C che abbia medesima distanza da A e B, tale punto C si trova sull'asse x' , asse delle ascisse del sistema di riferimento locale, e serve quindi a calcolarlo. Si sono quindi stimati i coefficienti angolari dei due asintoti rispetto al sistema di riferimento locale (x', y') che sono:

$$M1 = \tan(\tan^{-1}(m1) - \tan^{-1}(m3))$$

$$M2 = \tan(\tan^{-1}(m2) - \tan^{-1}(m3))$$

(3.13)

dove $m3$ è il coefficiente angolare di x' . Inoltre si sono trovate le coordinate del fuoco $F(Fx1, 0)$ rispetto a (x', y') come:

$$Fx1 = \frac{p}{\cos(\tan^{-1}(m3))}$$

(3.14)

dove p è stato scelto empiricamente uguale a 2 perché si è visto che tale valore non si discostava troppo dall'intersezione degli assi di riferimento. L'iperbole che stiamo costruendo è centrata nel sistema di riferimento locale quindi si è posta la direttrice

coincidente con l'asse y' . Per calcolare l'eccentricità dobbiamo fissare il vertice dell'iperbole $A'(a,0)$ e la sua proiezione su uno dei due asintoti $A1$; la distanza $\overline{A'A1} = \left| \frac{M2 * Fx1}{e+1} \right|$, dopo varie prove, è stata fissata pari a 3.2 quindi l'eccentricità è stata calcolata con la formula:

$$e = \frac{|M2| * Fx1}{\overline{A'A1}} - 1 \quad (3.15)$$

Infine si sono espresse la direttrice e le coordinate del fuoco rispetto al sistema di riferimento globale, esplicitando l'equazione dell'iperbole attraverso la definizione sopra descritta. L'eccentricità essendo un rapporto di distanze rimane costante nei due sistemi. La formula risultante dell'iperbole è un'equazione di secondo grado in t da cui se $m2 < 1$ si sceglie l'iperbole negativa, ossia con concavità verso il basso mentre se $m2 > 1$ si prende quella positiva.

L'approssimazione delle rette con l'iperbole è ottima per $m2$ nel range di interesse $[0.25; 1.5]$ esclusi i valori $0.99 < m2 < 1.01$ nel qual caso si considererà $x(t) = t$.

Il codice Matlab[®] relativo alla costruzione delle hrf di Handwerker è riportato in Cod.3.1a,b,c,d; mentre in Cod.3.2a,b,c vi è quello relativo alla costruzione dell'iperbole.

```
function h_ris=hrf_HW(ts,tf,delta1,tau1,a1,delta2,tau2,a2,delta,m2,tipo_hrf)

% Input fields:
% ts: frequenza di campionamento in secondi
% tf: periodo rest-active
% delta1: tempo di traslazione della gamma positiva y1(t) (default delta1=1.1)
% tau1: ampiezza della gamma positiva y1(t) (si tenga presente che l'ampiezza a
metà altezza FWHM=(15/13)*tau1; default tau1=5.2 )
% a1: altezza della gamma positiva y1(t) (default a1=0.167)
% delta2: tempo di traslazione della gamma negativa y2(t); per default posto a
delta1+tau1+tau1/2
% tau2: ampiezza della gamma negativa y2(t) (si tenga presente che l'ampiezza a
metà altezza FWHM=(15/13)*tau2; default tau2=1)
% a2: altezza della gamma negativa (default a2=0)
% delta: di quanto si vuole traslare la derivata di y1(t)+y2(t); (default delta=1)
% m2: coefficiente angolare della retta che aggiunge una maggiore dispersione
all'hrf (default m2=1)
% tipo_hrf: 'single_gamma' se si vuole che l'hrf sia uguale alla gamma positiva;
'two_gamma' se si vuole che l'hrf sia uguale alla somma della gamma positiva
y1(t) e della gamma negativa y2(t); 'two_gamma_der' se si vuole che l'hrf sia uguale
all'hrf precedente più la sua derivata temporale;
%
% Output fields:
% h_ris: l'hrf risultante dai parametri inseriti
% argument check
% se si inseriscono meno di 11 parametri il programma esce
if nargin ~= 11
    error('HRF:obj:hrf_HW','numero di parametri insufficiente');
end
fl=0; % flag delta2
fl1=0; % flag tau2
if ~ischar(tipo_hrf) ||...
    ~any(strcmpi(tipo_hrf, {'single_gamma','two_gamma','two_gamma_der'})),

    tipo_hrf = 'single_gamma';
else
    tipo_hrf = lower(tipo_hrf(:));
end
```

Cod.3. 1a: Codice relativo alla costruzione delle varie risposte emodinamiche di Handwerker.

```
if ~isa(m2, 'double') || ...
numel(m2) ~= 1 || ...
isinf(m2) || ...
isnan(m2) || m2==0
m2 = 1;
end
if ~isa(delta, 'double') || ...
numel(delta) ~= 1 || ...
isinf(delta) || ...
isnan(delta)
delta = 1;
end
if ~isa(a2, 'double') || ...
numel(a2) ~= 1 || ...
isinf(a2) || ...
isnan(a2) || ...
a2 < 0
a2 = 0;
end
if ~isa(tau2, 'double') || ...
numel(tau2) ~= 1 || ...
isnan(tau2) || ...
tau2 <= 0
fl1=1;
end
if ~isa(delta2, 'double') || ...
numel(delta2) ~= 1 || ...
isinf(delta2) || ...
isnan(delta2)

fl=1;
end
if ~isa(a1, 'double') || ...
numel(a1) ~= 1 || ...
isinf(a1) || ...
isnan(a1) || ...
a1 <= 0
a1 = 0.167;
end
if ~isa(tau1, 'double') || ...
numel(tau1) ~= 1 || ...
isinf(tau1) || ...
isnan(tau1) || ...
tau1 <= 0
tau1 = 5.2;
end
if ~isa(delta1, 'double') || ...
numel(delta1) ~= 1 || ...
isinf(delta1) || ...
isnan(delta1) || ...
delta1 < 0
delta1 = 1.1;
end
```

Cod.3.1b: Codice relativo alla costruzione delle varie risposte emodinamiche di Handwerker.

```

if ~isa(tf, 'double') || ...
    numel(tf) ~= 1 || ...
    isinf(tf) || ...
    isnan(tf)
    error('HRF: obj: hrf_HW', 'periodo rest-active mancante o non corretto');
end
if ~isa(ts, 'double') || ...
    numel(ts) ~= 1 || ...
    isinf(ts) || ...
    isnan(ts)
    error('HRF: obj: hrf_HW', 'tempo di campionamento TR mancante o non corretto');
end

if fl==1 || delta2 <= delta1+tau1,
    delta2=delta1+tau1+tau1/2;
end
if fl1==1,
    tau2=tau1/2;
end
if a1<2*a2,
    a1=2*a2;
end

t=(0:ts:tf-ts); % vettore dei campioni dell'hrf
%-----Relazioni della curva y1(t)-----
nA1=a1*exp(1);
A1=nA1*tau1; % A1&tau1
tmax=delta1+tau1;
%-----
%-----Parametri di y2(t)=curva negativa-----
nA2=a2*exp(1);
A2=-nA2*tau2; %A2&tau2
%-----
%--vettore che azzerà i parametri poco significativi di y1---
dd1=(0:ts:delta1);
d1=[zeros(1,length(dd1)) ones(1,length(t)-length(dd1))];
%-----
% se m2=1 STRADA DELL' HRF CON x(t) VETTORE t
if m2~=1,% STRADA DELL' HRF CON x(t)=yip IPERBOLE
    [yip]=ip(t,tmax,m2); %calcolo dell'iperbole
    t=yip;
end
% y1(t) curva positiva
y1=d1.*A1.*((t-delta1)./tau1).^2.*(exp(-(t-delta1)./tau1).^2))./tau1;
%--vettore che azzerà i parametri poco significativi di y2---
dd2=(0:ts:delta2);
d2=[zeros(1,length(dd2)) ones(1,length(t)-length(dd2))];
%-----
% y2(t)=curva negativa
y2=d2.*A2.*((t-delta2)./tau2).^2.*(exp(-(t-delta2)./tau2).^2))./tau2;
h=y1+y2; % hrf two-gamma
%hder1=d1.*diff(h)./ts;

```

Cod.3.1c: Codice relativo alla costruzione delle varie risposte emodinamiche di Handwerker.

```

%---Derivate-----
if m2==1,
% Derivata di y1(t)
y1der=d1.*(2*(A1/(tau1^3)).*(t-delta1).*exp(-((t-delta1)./tau1).^2).*(1-((t-
delta1)./tau1).^2));
% Derivata di y2(t)
y2der=d2.*(2*(A2/(tau2^3)).*(t-delta2).*exp(-((t-delta2)./tau2).^2).*(1-((t-
delta2)./tau2).^2));
%-----
else
%---Derivate-----
dip=[0 diff(t)./ts];% dx(t)/dt dove x(t) è l'iperbole
% Derivata di y1(t)
y1der=d1.*(2*(A1/(tau1^3)).*dip.*(t-delta1).*exp(-((t-delta1)./tau1).^2).*(1-((t-
delta1)./tau1).^2));
% Derivata di y2(t)
y2der=d2.*(2*(A2/(tau2^3)).*dip.*(t-delta2).*exp(-((t-delta2)./tau2).^2).*(1-((t-
delta2)./tau2).^2));
%-----
end
hder=y1der+y2der;% derivata dell'hrf
h_f=h-hder.*delta;% hrf finale

switch (tipo_hrf)

% single-gamma HRF
case {'single_gamma'}
    h_ris=y1;

% two-gamma HRF
case {'two_gamma'}
    h_ris=h;

% two-gamma+derivata temporale HRF
case {'two_gamma_der'}
    h_ris=h_f;
end

```

Cod. 3.1d: Codice relativo alla costruzione delle varie risposte emodinamiche di Handwerker.

```

function yip=ip(x,tmax,m2)

% Input fields:
% x: asse delle ascisse
% tmax: ascissa d'intersezione tra gli asintoti dell'iperbole rispetto al sistema globale
% m2: coefficiente angolare del secondo asintoto, range 0.25<=m2<=1.5
%
% Output fields:
% yip: iperbole asintotica alle due rette di coefficienti m1, m2; utilizzata per rendere
continua la      derivata dell'hrf in corrispondenza dell'intersezione tra le due rette.

```

Cod.3.2a: Codice relativo alla costruzione dell'iperbole.

```

% argument check
if nargin < 3 || ...
    ~isa(m2, 'double') || ...
    numel(m2) ~= 1 || ...
    isinf(m2) || ...
    isnan(m2) || ...
    m2>1.5 || m2<0.25

    disp(['m2 non corretto o fuori range x= ' num2str(x) ', tmax= ' num2str(tmax) ',
m2= ' num2str(m2)'])
    % m2=1;
    return
end
if nargin < 2 || ...
    ~isa(tmax, 'double') || ...
    numel(tmax) ~= 1 || ...
    isinf(tmax) || ...
    isnan(tmax)
    error('function: ip', 'punto di intersezione tmax mancante o non corretto');
end
if nargin < 1 || ...
    ~isa(x, 'double') || ...
    numel(x) < 3 || ...
    sum(isinf(x))~=0 || ...
    sum(isnan(x))~=0
    error('function: ip', 'asse x delle ascisse non corretto o numero di campioni
insufficiente');
end

m1=1; % coefficiente angolare del primo asintoto che è fisso
q1=0; % intercetta del primo asintoto fissa
q2=tmax*(m1-m2)+q1; % intercetta del secondo asintoto
% asintoti
y1=m1.*x+q1;
y2=(m2.*x)+q2;
% punti equidistanti dall'origine O1(tmax,ytmax) sui 2 asintoti A=(ax,ay) B=(bx,by)
ytmax=m1*tmax+q1;
ax=tmax-3;
ay=m1*ax+q1;
% distanza O1A al quadrato
O1A=(tmax-ax)^2+(ytmax-ay)^2;
% cerco il punto B tale che O1A=O1B e che bx>tmax
% la soluzione dell'equazione di secondo grado di bx è B+-sqrt(B^2-C)
k=(1+m1^2)/(1+m2^2);
bx1=tmax+3*sqrt(k);
bx2=tmax-3*sqrt(k);
if bx1>tmax
    bx=bx1;
else
    bx=bx2;
end
by=(m2*bx)+q2;

```

Cod.3.2b: Codice relativo alla costruzione dell'iperbole.

```

% punto C=(cx,cy) sull'asse x' trovato considerando le distanze CB=CA
cx=tmax+3;
cy=((cx-bx)^2-(cx-ax)^2+by^2-ay^2)/(2*by-2*ay);
m3=(ytmax-cy)/(tmax-cx);
q3=cy-m3*cx;

%asse x'
y3=m3.*x+q3;
% Asintoti rispetto al sistema di riferimento locale y'x'
%1) y'=M1*x'
M1=tan(atan(m1)-atan(m3));
%2) y'=M2*x'
M2=tan(atan(m2)-atan(m3));

% Fuoco in coordinate del sistema di riferimento locale (fx1,0) si è preso un valore, 2
che non si discosti troppo dall'intersezione degli assi di riferimento
fx1=2/cos(atan(m3));
% la direttrice la considero coincidente con l'asse y', quindi per definizione e=(fx1-
a)/a, con vertice dell'iperbole in A(a,0). AA1 è la distanza tra il vertice
dell'iperbole e la sua proiezione su uno dei due asintoti
AA1=3.2; % fissato dopo varie prove fissando empiricamente l'eccentricità
AA1=abs(M2*fx1/(e+1))
e=(abs(M2)*fx1/AA1)-1; % eccentricità dell'iperbole

%asse y'
q4=tmax*(m1+(1/m3));
y4=-(1/m3).*x+q4;

%direttrice rispetto al sistema di riferimento globale
dx=tmax;
dy=m3*dx+q3;
q5=dy+(dx/m3);
y5=-(1/m3).*x+q5;
%fuoco F=(fx,fy) rispetto al sistema di riferimento globale
fx=tmax+2;
fy=m3*fx+q3;

%trova l'iperbole (1-A)[y^2+y*D/(1-A)^2-B/(1-A)] in coordinate rispetto al sistema
globale
A=sqrt((1/m3)^2+1)/e^2;
B=(-x.^2./m3^2)-q5^2+((2.*x.*q5)./m3)+(fy^2*A)+A.*(x-fx).^2;
D=2*fy*A+(2.*x)./m3-2*q5;
if m2==1
    yip=x+q2;
end
yip1=(-D./(1-A))+sqrt((D.^2/(1-A)^2)+4.*B/(1-A))./2;
yip2=(-D./(1-A))-sqrt((D.^2/(1-A)^2)+4.*B/(1-A))./2;
if m2>1
    yip=yip1;
elseif m2<1
    yip=yip2;
end
yip=(yip+abs(yip))./2; % pongo a zero la parte negativa dell'iperbole

```

Cod.3.2c: Codice relativo alla costruzione dell'iperbole.

Capitolo 4

Generazione di mappe di attivazione: hrf prefissata e uguale per ogni soggetto

L'analisi statistica dei dati fMRI fa delle deduzioni basandosi su modelli di dati, che contengono una grande quantità di errore casuale, che in certi casi arriva a raggiungere anche il 99% del segnale BOLD. Il tipo di analisi può essere sviluppata costruendo un modello che tenga conto che la risposta BOLD dipende dallo stimolo; inoltre deve contenere una componente di errore casuale, che spiega come variano le osservazioni, anche se l'esperimento è ripetuto sullo stesso soggetto sotto le stesse condizioni. In questa tesi utilizzeremo l'analisi statistica sia per stimare i parametri del modello, che per rilevare le mappe di attivazione. I metodi statistici più utilizzati a questo scopo sono: l'Analisi di correlazione, il General Linear Model (GLM), l'Analisi di Fourier e l'Independent Component Analysis (ICA). Nel presente elaborato verrà utilizzato il *General Linear Model*.

4.1 General Linear Model, GLM

Costruzione del modello

Vediamo come la risposta BOLD dipende da uno stimolo esterno. Il tempo verrà indicato con t e lo stimolo esterno già introdotto nel capitolo 2 con $s(t)$. La risposta BOLD di un particolare voxel, $x(t)$, di solito si verifica tra 3 e 10 s dopo lo stimolo, il picco è a circa 6 s; questo ritardo è modellato con la funzione della risposta emodinamica $h(t)$ che pesa il valore dello stimolo con una convoluzione come segue:

$$x(t) = \int_0^{\infty} h(u)s(t-u)du$$

(4.1)

La convoluzione risultante $x(t)$ viene poi campionata in N volumi di acquisizione fMRI nei tempi t_1, \dots, t_N e dà la risposta $x_i = x(t_i)$ al volume i -esimo.

In generale possiamo avere stimoli diversi $s_1(t), \dots, s_M(t)$ a cui corrisponderanno attraverso la convoluzione m risposte diverse $x_1(t), \dots, x_M(t)$.

Inoltre i valori $x_1(t), \dots, x_M(t)$ vengono poi moltiplicati per i diversi parametri β_1, \dots, β_M e tali prodotti sono infine sommati insieme per ottenere la risposta finale. I parametri β_1, \dots, β_M sono incogniti e rappresentano l'ampiezza con cui si adatta il modello relativo ai vari task. Questa modellizzazione lineare è stata proposta da Friston et al. nel 1995:

$$x_{i1}\beta_1 + \dots + x_{iM}\beta_M; \quad [10]$$

(4.2)

I modelli statistici sono composti normalmente dagli effetti fissi e dall'errore random. Gli effetti fissi sono le parti di un modello che non variano se l'esperimento viene ripetuto; essi colgono l'intrinseca verità scientifica che si vuole scoprire. L'errore random è la parte che varia ogni volta che vengono ottenuti nuovi dati. Questo errore è molto importante per due ragioni la prima, ci dice quanto migliora la stima dell'effetto dello stimolo (effetto fisso) e la seconda ci fornisce una via per stimare l'errore nell'effetto questo ci permette di confrontare l'effetto con il suo errore random e selezionare quei voxel dove l'effetto è molto più grande del suo errore random, cioè i voxel con elevato rapporto segnale rumore (SNR).⁷

Formulazione matriciale e stima dei parametri

Aggiungendo l'errore random ε_i si ottengono i dati fMRI osservati Y_i dalla seguente formula:

$$Y_i = x_{i1}\beta_1 + \dots + x_{iM}\beta_M + \varepsilon_i \quad (4.3)$$

in notazione matriciale:

⁷ SNR sta per Signal to Noise Ratio.

$$\mathbf{Y} = \mathbf{X} * \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

con

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_N \end{pmatrix}; \quad \mathbf{X} = \begin{pmatrix} X_{11} & \cdots & X_{1M} \\ \vdots & \ddots & \vdots \\ X_{N1} & \cdots & X_{NM} \end{pmatrix}; \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_M \end{pmatrix}; \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_N \end{pmatrix};$$

(4.4)

dove N è il numero di volumi o osservazioni e M è il numero dei diversi stimoli che vengono modellati. \mathbf{X} è detta *design matrix* e dipende dalla forma attesa della risposta BOLD ai vari istanti temporali. Il modello \mathbf{X} e i dati \mathbf{Y} dei singoli voxel sono noti mentre il vettore dei parametri $\boldsymbol{\beta}$ è incognito ed è stimato attraverso uno stimatore ai minimi quadrati:

$$\hat{\boldsymbol{\beta}} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \tilde{\mathbf{Y}} \quad \text{dove} \quad \tilde{\mathbf{X}} = \mathbf{K} \mathbf{W} \mathbf{X}; \quad \tilde{\mathbf{Y}} = \mathbf{K} \mathbf{W} \mathbf{Y} \quad (4.5)$$

Le matrici \mathbf{K} e \mathbf{W} sono di seguito specificate, in particolare \mathbf{W} verrà posta inizialmente a \mathbf{I} per i motivi che vedremo. L'unica variabile incognita rimanente è la stima del rumore $\boldsymbol{\varepsilon}$ o residuo che per la 4.4 è data da:

$$\hat{\boldsymbol{\varepsilon}} = \tilde{\mathbf{Y}} - \tilde{\mathbf{X}} * \hat{\boldsymbol{\beta}}$$

(4.6)

Matrice di filtraggio temporale \mathbf{K}

La matrice \mathbf{K} è una matrice di filtraggio ad alta frequenza, è costruita attraverso la *discrete cosine transform DCT*:

$$K(n, t) = \begin{cases} \frac{1}{\sqrt{N}} & t = 0 \\ \sqrt{\frac{2}{N}} \cos \frac{\pi(2n+1)t}{2N} & 1 < t < \frac{2(N * TR)}{cutoff} + 1 \end{cases}$$

(4.7)

questo filtro passa-alto va a moltiplicare la design matrix e rimuove le componenti a bassa frequenza indesiderate. In particolare taglia le frequenze al di sotto di $1/cutoff$ nell'implementazione si è scelto di mantenere il $cutoff=128$ s, ossia il valore di default di SPM.[9],[10]

L'obiettivo di tale filtraggio è preservare l'informazione riguardante i cambiamenti nel dato che avvengono alla frequenza del task applicato e minimizzare quelli che avvengono ad altre frequenze. Inoltre l'uso di un filtro temporale può migliorare sostanzialmente la qualità del dato fMRI aumentando SNR.

Nel seguito indicheremo $\hat{\mathbf{X}}=\mathbf{K}\mathbf{X}$ e $\hat{\mathbf{Y}}=\mathbf{K}\mathbf{Y}$.

Costruzione della matrice di pesatura \mathbf{W}

Uno dei metodi più semplici per la stima della matrice dei pesi \mathbf{W} è il metodo di Cochrane-Orcutt che stima per primo il vettore $\boldsymbol{\beta}$ con i minimi quadrati partendo dai dati originali non smoothed e pone \mathbf{W} uguale alla matrice identità \mathbf{I} . Questo stimatore è sempre unbiased anche se forse non è più preciso, ma assicura che i residui contengano solo errore e non segnale, dovuto al fatto che l'aspettazione dei residui è zero. Inoltre la struttura di correlazione dei residui è strettamente approssimata dalla matrice \mathbf{V} che vogliamo stimare.[10]

L'errore di misura $\boldsymbol{\varepsilon}$ inizialmente è stato considerato scorrelato, additivo, con distribuzione normale a media nulla e varianza $\sigma^2\mathbf{I}$, cioè $\boldsymbol{\varepsilon}\sim\mathcal{N}(0, \sigma^2\mathbf{I})$. [10]

In realtà le osservazioni tendono ad essere correlate nel tempo; tale fenomeno è noto come *autocorrelazione temporale* o *smoothness* e rappresenta l'influenza che i precedenti punti temporali, modificati dall'errore random, hanno sul punto temporale corrente. Questo disturbo è causato in particolare dalla pulsazione cardiaca e dalla respirazione.[10]

Una modellizzazione efficace della correlazione temporale è data dal modello autoregressivo del primo ordine, AR(1). In questo modo l'errore del precedente punto temporale viene combinato con un nuovo termine di errore per produrre l'errore temporale corrente:

$$\varepsilon_i = \rho\varepsilon_{i-1} + \chi_{i1} \quad (4.8)$$

dove il sistema di equazioni di Yule Walker relativo è stabile se il parametro ρ soddisfa $|\rho| < 1$.

Inoltre χ_{i1} è rumore bianco, una sequenza di variabili random indipendenti, normalmente distribuite con media zero e deviazione standard σ_1 , $\chi_{i1}\sim\mathcal{N}(0, \sigma_1^2)$. Con

un tale modello, la correlazione temporale decade esponenzialmente all'aumentare del ritardo.[10],[19]

Il filtraggio ottenuto con la pre-moltiplicazione attraverso la matrice \mathbf{K} compensa il bias provocato dalla posizione di assumere ρ costante in tutto il cervello, ossia fissato alla media di tale parametro su tutti i voxel del cervello.

In seguito assumeremo che la distribuzione di $\boldsymbol{\varepsilon}$ sia una normale multivariata in modo da poter descrivere l'intero modello lineare come:

$$\mathbf{Y}=\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}; \text{ con } \boldsymbol{\varepsilon}\sim N(0,\sigma^2\mathbf{V}) \quad (4.9)$$

dove $\mathbf{V}\sigma^2$ è la matrice di covarianza dell'errore $\boldsymbol{\varepsilon}$, σ^2 è uno scalare incognito e la matrice \mathbf{V} per un modello autoregressivo del primo ordine è calcolata come segue:

$$\mathbf{V} = \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{n-1} \\ \rho & 1 & \rho & \dots & \rho^{n-2} \\ \rho^2 & \rho & 1 & \dots & \rho^{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \dots & 1 \end{pmatrix}; \quad \sigma^2 = \frac{\sigma_1^2}{1-\rho^2} \quad [10] \quad (4.10)$$

con σ_1 e ρ note e calcolate attraverso il modello AR(1) visto precedentemente. \mathbf{V} è una matrice di Toeplitz, cioè simmetrica e con le righe uguali alle corrispondenti colonne.[2],[10]

La migliore stima del vettore $\boldsymbol{\beta}$ si può ottenere, con il teorema di Gauss-Markov, scegliendo \mathbf{W} in modo che la varianza degli errori sia proporzionale alla matrice identità, equivalente allo 'sbiancamento degli errori'. Questo processo rimuove ogni smoothness temporale nei dati se causato da pre-filtraggio o da smoothness intrinseco degli errori random. Una possibile soluzione è quella di utilizzare la fattorizzazione di Cholesky su \mathbf{V} da cui successivamente si può calcolare la matrice dei pesi \mathbf{W} :

$$\mathbf{V}=\mathbf{H}^T\mathbf{H}; \quad \mathbf{W}=(\mathbf{H}^T)^{-1}; \quad \mathbf{W}\mathbf{V}\mathbf{W}^T=\mathbf{I}; \quad [19]$$

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -\rho R & R & 0 & \dots & 0 \\ 0 & -\rho R & R & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -\rho R & R \end{pmatrix} \quad R = \frac{1}{\sqrt{1-\rho^2}} \quad [10]$$

(4.11)

Ora attraverso la 4.5 e la 4.6 si stimano nuovamente il vettore $\boldsymbol{\beta}$ e il vettore $\boldsymbol{\varepsilon}$ con la matrice dei pesi \mathbf{W} appena calcolata e successivamente si stima a posteriori la costante di proporzionalità della varianza dell'errore, $\hat{\sigma}^2$, utilizzando la seguente relazione [20]:

$$\hat{\sigma}^2 = \frac{\text{WRSS}}{N - M};$$

(4.12)

dove WRSS è la somma dei residui al quadrato pesati ed è data dalla formula:

$$\text{WRSS} = (\hat{\mathbf{Y}} - \hat{\mathbf{X}} * \hat{\boldsymbol{\beta}})^T \mathbf{V}^{-1} (\hat{\mathbf{Y}} - \hat{\mathbf{X}} * \hat{\boldsymbol{\beta}}) = \hat{\boldsymbol{\varepsilon}}^T \mathbf{V}^{-1} \hat{\boldsymbol{\varepsilon}}. \quad [20],[2]$$

(4.13)

Infine si trova la covarianza della stima dei parametri, che è un indice di precisione delle stime:

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \hat{\sigma}^2 \hat{\mathbf{X}}^{-1} \mathbf{V} \hat{\mathbf{X}}^{-T}$$

(4.14)

Scelta tra modelli in competizione

Nel confrontare la capacità dei modelli di descrivere i dati sperimentali bisogna guardare non solo il *fit* del modello, cioè come essi predicono i dati, ma anche la precisione delle stime dei parametri ottenuti.[22] Per poter scegliere tra vari modelli in competizione nel presente lavoro si è utilizzato il criterio di Akaike che si applica a modelli lineari con cui viene selezionato il metodo associato all'indice più basso. In particolare è stato calcolato tale indice per ogni voxel come segue:

$$\text{AIC} = N * \log(\text{WRSS}) + 2 * M + \frac{2 * M * (M+1)}{(N-M-1)} \quad [21]$$

(4.15)

dove N sono il numero di dati e M il numero di parametri. Come si deduce dalla formula l'indice di Akaike dipende oltre che dal numero di dati anche dalla complessità del modello. Infine si sono calcolati i coefficienti di variazione del singolo parametro β_i stimato come:

$$CV(\hat{\beta}_i) = \frac{\sqrt{\text{Var}(\hat{\beta})}}{\hat{\beta}_i} * 100 \quad [20]$$

(4.16)

Per poter applicare una prima selezione dei voxel attivi, si prende come soglia la mediana di tutti i voxel con $CV(\hat{\beta}_1) > 0$, selezionando solo quei voxel con $0 < CV(\hat{\beta}_1) < \text{soglia}$. Questa preselezione è l'insieme su cui opera la successiva elaborazione statistica del T-test.

T-test

Le particolari differenze dei parametri β sono specificate dal vettore contrasto \mathbf{c} , che ha la stessa lunghezza di β . Il più semplice contrasto usato nel codice di questa tesi coinvolge una sola variabile esplicativa, ad esempio il test di attivazione nella condizione event confrontato con il rest, il vettore contrasto è $\mathbf{c}^T = (1 \ 0)$. Questo significa che la stima dell'effetto è semplicemente $\mathbf{c}^T \hat{\beta} = \hat{\beta}_1$.

L'ipotesi nulla H_0 assunta come base di questo test è $\hat{\beta}_1 = \hat{\beta}_2$ cioè rileva i voxel non attivi in cui vi sono le stesse condizioni event-rest.

Il T statistico è dato da:

$$T = \frac{\mathbf{c}^T \hat{\beta}}{\sigma \sqrt{\mathbf{c}^T (\hat{\mathbf{X}}^T \hat{\mathbf{X}})^{-1} \mathbf{c}}}$$

(4.17)

dove il numeratore rappresenta le stime dei parametri pesate per il contrasto e il denominatore è la deviazione standard del numeratore. Poiché nel nostro caso i voxel di interesse sono quelli attivi, ossia con $\hat{\beta}_1 \neq \hat{\beta}_2$, solo questi voxel verranno utilizzati per le elaborazioni successive. A questo punto scegliamo il livello di significatività $p=0.05$, tale soglia esclude i falsi positivi con probabilità 0.95 questo significa che solo il 5% delle parti non attivate del cervello mostreranno falsi positivi. Visto l'alto numero di voxel per fetta, 4096, la percentuale del 5% comporta un discreto numero

di voxel falsi positivi quindi si ricorre a un'ulteriore selezione attraverso la correzione di Bonferroni.

Correzione di Bonferroni

La correzione di Bonferroni controlla i *falsi positivi*, ossia la non classificazione di voxel che sono attivi, mentre tende a risultare troppo conservativa in quanto incrementa i *falsi negativi*, cioè la non classificazione di zone che in realtà sono attive. La disuguaglianza di Bonferroni è applicata ad ogni voxel all'interno della maschera specifica del singolo soggetto ed è data da:

$$a_T \leq k * a \quad (4.18)$$

dove a_T è la probabilità che vi sia almeno un voxel in cui viene rifiutata l'ipotesi nulla H_0 , k è il numero di voxel di una fetta che si trovano all'interno della maschera, cioè fanno parte del cervello di quello specifico soggetto, e a è la soglia di Bonferroni. In particolare la soglia utilizzata nella presente tesi è data da $0.05/k$, quindi $a_T=5\%$.

Poiché il nostro obiettivo è trovare l'insieme di funzioni base il cui modello dia il miglior *fit* possibile dei dati fMRI, a questo proposito si è implementato in questa tesi anche un metodo simile a quello utilizzato da SPM in cui la risposta emodinamica viene modellata come combinazione lineare di un insieme di funzioni base.

4.2 Applicazione dei GLM al data set

Nella generazione delle mappe di attivazione è estremamente rilevante la scelta iniziale della risposta emodinamica $h(t)$ è possibile assumere un'hrf prefissata, ossia fissare una risposta emodinamica $h(t)$ uguale per tutti i soggetti. In letteratura le hrf principali sono di due tipologie: la single-gamma e la two-gamma a queste sono state aggiunte altre due varianti in cui si è tenuto conto anche della derivata temporale e della derivata dispersiva. A seconda della scelta della $h(t)$ si avranno quattro metodi:

Metodo 7: hrf single-gamma di Handwerker

In tale metodo il modello che si utilizza nel GLM, viene costruito ponendo come $h(t)$ la single-gamma HW vista nel paragrafo 3.2.2. Il vettore dei parametri relativo sarà $\beta=[\beta_1 \ \beta_2]^T$ dove β_1 modella l'event e β_2 il rest. La formulazione matriciale del GLM relativa a questo metodo è:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} x_{11} & 1 \\ x_{21} & 1 \\ \vdots & \vdots \\ x_{N1} & 1 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix} \quad (4.19)$$

$$\mathbf{Y} = \mathbf{X} * \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Per il calcolo dell'indice di AIC nella eq. 4.15 si avrà $M=2$; nel T-test il vettore contrasto utilizzato è $\mathbf{c}=[1 \ 0]^T$. [18]

Metodo 8: hrf two-gamma di Handwerker

Tale metodo è del tutto simile al metodo precedente l'unica differenza sta nella hrf utilizzata nel modello che in questo caso è la hrf two-gamma HW vista nel paragrafo 3.2.2. [18]

Metodo 9: Gamma Function Temporal (GFT)

Con questo metodo il modello è stato costruito con la seguente formulazione matriciale:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} x_{11} & 1 & x_{13} \\ x_{21} & 1 & x_{23} \\ \vdots & \vdots & \vdots \\ x_{N1} & 1 & x_{N3} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

$$\mathbf{Y} = \mathbf{X} * \boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (4.20)$$

dove \mathbf{Y} sono i dati di un singolo voxel e N sono i volumi o campioni acquisiti per quel voxel. La matrice \mathbf{X} è il modello che descrive la risposta emodinamica, è di dimensione $N \times 3$; la prima colonna è data dalla convoluzione tra l'hrf two-gamma HW con il protocollo sperimentale visto nel capitolo 2, la seconda colonna rappresenta la condizione di rest e il valore 1 nel codice MATLAB[®] implementato è stato sostituito con il valore massimo del modello. La colonna tre è data dalla

convoluzione della derivata temporale della two-gamma HW, formula 3.10, con il protocollo sperimentale. In questo caso il vettore dei parametri da stimare β ha tre componenti, quindi il valore dell'indice di AIC calcolato con la eq.4.15 avrà $M=3$; infine il vettore del contrasto utilizzato nel T-test è del tipo $c=[1 \ 0 \ 0]^T$. [18]

Metodo 10: Gamma Function Temporal Dispersion (GFTD)

In questo caso la formulazione matriciale è data da:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} x_{11} & 1 & x_{13} & x_{14} \\ x_{21} & 1 & x_{23} & x_{24} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & 1 & x_{N3} & x_{N4} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

$$Y = X * \beta + \varepsilon$$

(4.21)

nella matrice X è stata aggiunta una quarta colonna che rappresenta la convoluzione tra la derivata dispersiva della two-gamma di HW e il protocollo sperimentale. La derivata dispersiva è data da, $\frac{dy_1}{d\tau_1} + \frac{dy_2}{d\tau_2}$ dove $\frac{dy_1}{d\tau_1}$ è la 3.11 e $\frac{dy_2}{d\tau_2}$ si ottiene dalla stessa formula sostituendone gli opportuni parametri. Poiché il vettore β è di rango 4, l'indice di AIC ha $M=4$ e il vettore contrasto è dello stesso rango cioè ad esempio $c=[1 \ 0 \ 0 \ 0]^T$. [18]

Capitolo 5

Generazione di mappe di attivazione: hrf specifica per ogni soggetto

La risposta emodinamica varia sia da soggetto a soggetto sia tra le aree cerebrali dello stesso paziente. In questo contesto verrà utilizzata un'hrf media rappresentativa delle zone cerebrali interessate dallo stimolo di un determinato individuo. In questo capitolo vedremo come si è implementato il codice Matlab[®] per la costruzione dell'hrf relativa al singolo soggetto che viene infine utilizzata per la generazione delle mappe di attivazione.

5.1 Metodi per la generazione di mappe di attivazione con hrf specifica per ogni soggetto

In questa tesi si sono sviluppati quattro metodi relativi alla generazione di mappe di attivazione specifiche per soggetto. Tali metodi si sviluppano, come si intuisce dai diagrammi di flusso qui proposti, in quattro fasi operative. La *prima fase* serve a isolare un insieme di voxel sicuramente attivi, considerati voxel campione per quel soggetto. Essa si suddivide nelle due sottofasi di *selezione* e *raggruppamento*. La *selezione* ha lo scopo di candidare un numero di voxel come attivi; mentre il *raggruppamento* conferma i voxel di *selezione* che si presentano in gruppo ed elimina gli altri. Nella *seconda fase* ogni voxel selezionato viene processato attraverso una stima ai minimi quadrati non lineari il cui risultato sono i parametri della specifica risposta emodinamica e i relativi coefficienti di variazione, CV. La *terza fase*, utilizza i dati dell'elaborazione precedente, e sceglie i voxel più rappresentativi attraverso opportune soglie applicate ai parametri e ai relativi CV. In

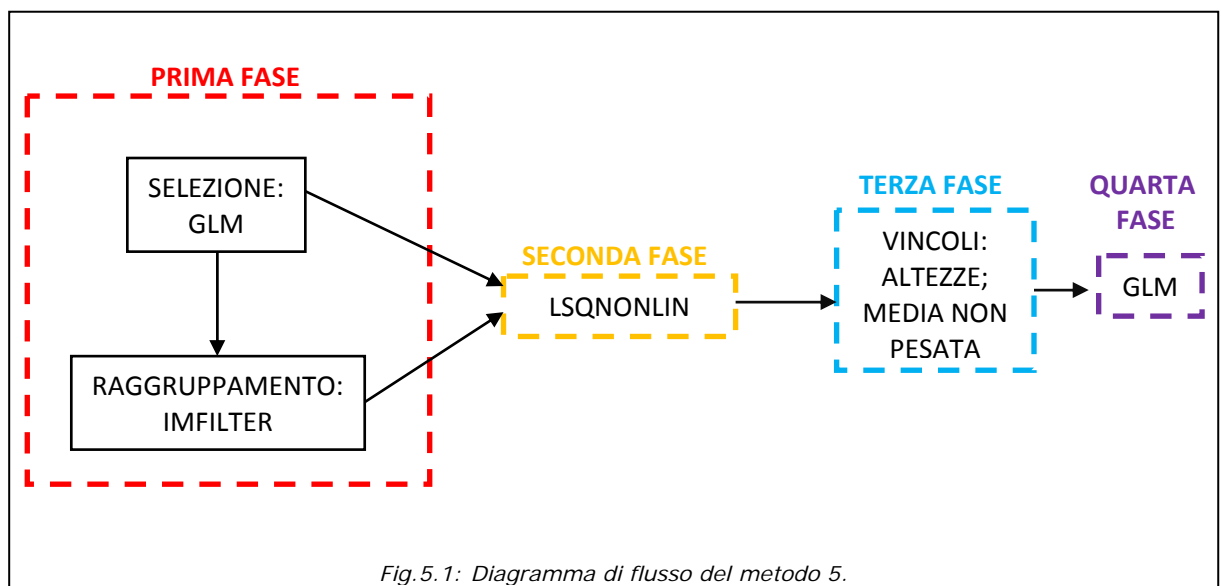
seguito, facendo la media dei parametri, si ricava la risposta emodinamica specifica per quel soggetto. L'hrf così ottenuta verrà utilizzata nella *quarta fase* dal GLM per creare la mappa di attivazione relativa. In questo caso il GLM è come quello utilizzato nei metodi 7 e 8 del paragrafo 4.2, ossia con vettore $\beta = [\beta_1 \beta_2]$.

Analizziamo ora i quattro metodi implementati:

Metodo 5: hrf con media dei parametri

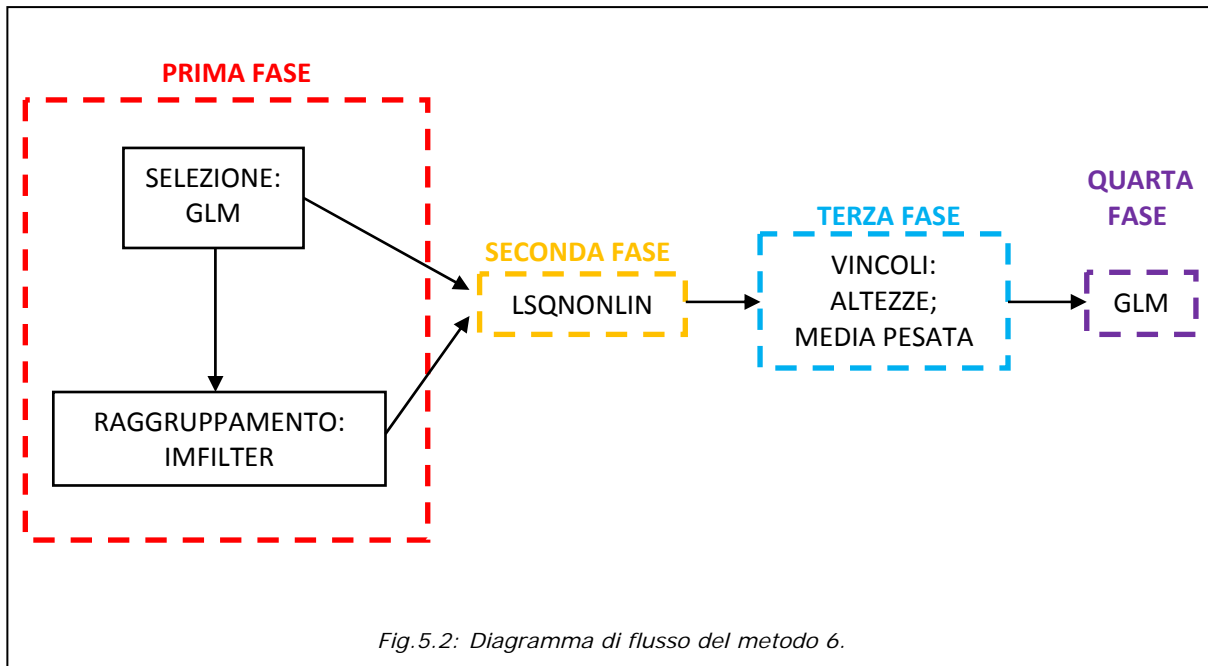
Il primo metodo è rappresentato in figura 5.1. Si può notare che nella prima fase la selezione dei voxel attivi viene fatta attraverso l'applicazione del metodo GLM visto nel capitolo 4 in cui l'hrf usata è l'hrf two-gamma HW e il raggruppamento è svolto attraverso la function *raggruppa_vox* che utilizza *imfilter* di Matlab[®] di cui parleremo nel corso di questo capitolo. Nel caso in cui l'uscita del raggruppamento elimini tutti i voxel sarà l'uscita del GLM stesso ad essere direttamente processata nella seconda fase dalla function di Matlab[®] *lsqnonlin*. Ai voxel processati con i minimi quadrati non lineari viene poi applicata una soglia sulle altezze a_1 e a_2 e sui relativi coefficienti di variazione, per dettagli si veda il paragrafo 5.4.1. Se non vi è alcun voxel che supera tale soglia si ripete l'elaborazione della *seconda fase* sui voxel della sottofase *selezione*.

Infine per trovare la risposta emodinamica specifica del soggetto si applica la media sui parametri dell'hrf_{fin} e tale curva risultante viene poi utilizzata nell'analisi del GLM che selezionerà le aree di attivazione del soggetto.



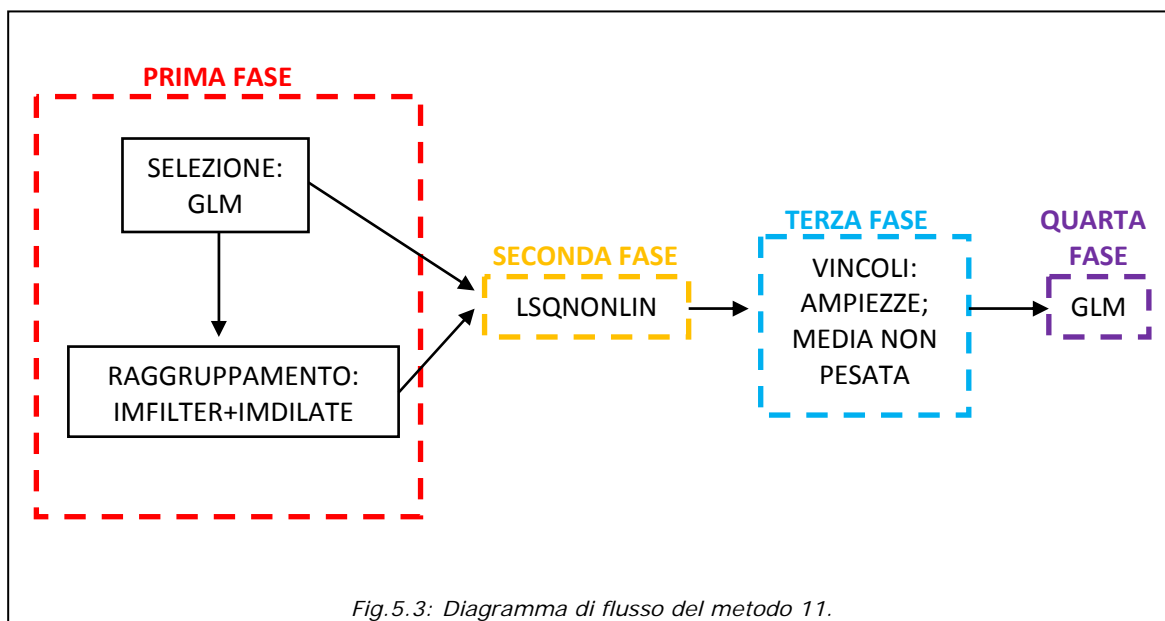
Metodo 6: hrf con media pesata dei parametri

Il metodo 6 è del tutto simile al 5 tranne nella terza fase in cui invece di calcolare semplicemente la media dei parametri per trovare l'hrf specifica del soggetto, si trova una media pesata sui coefficienti di variazione dei singoli parametri. Il diagramma di flusso corrispondente è rappresentato in figura 5.2.



Metodo 11: hrf con media dei parametri modificata

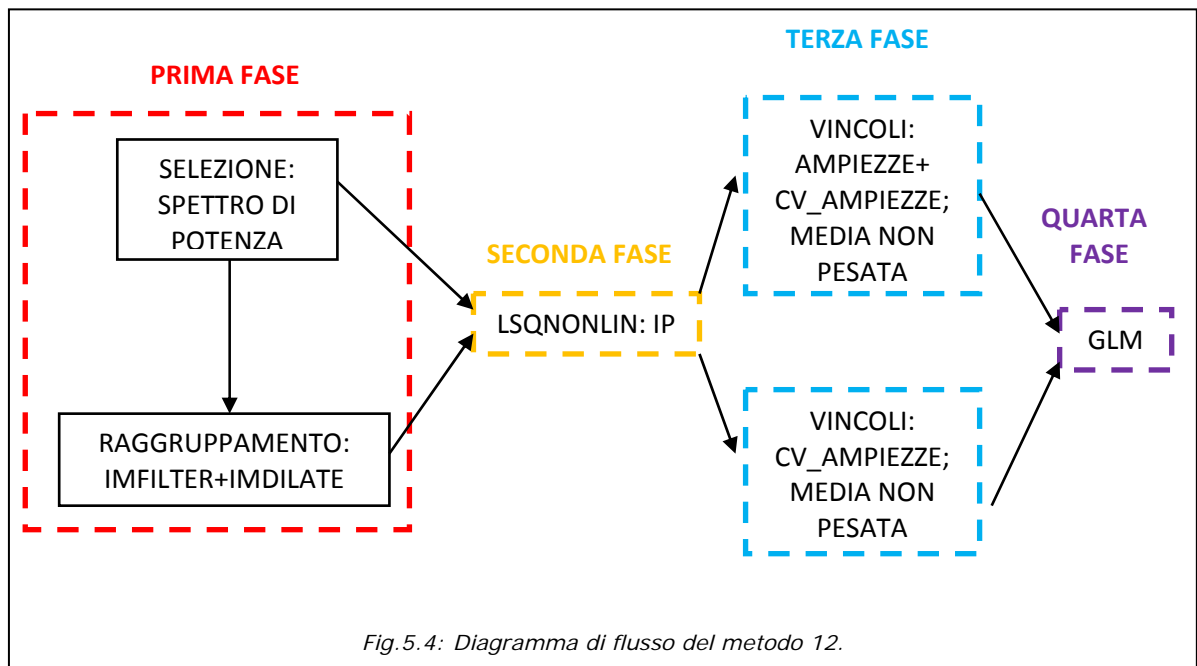
Si tratta di una variante del metodo 6 in cui nella prima fase il



raggruppamento utilizza oltre al filtraggio spaziale anche un'operazione morfologica rendendo l'insieme selezionato più conservativo (sottoparagrafo 5.2.2). Un'altra differenza rispetto al primo metodo è anche nella terza fase dove il tipo di vincolo non è più relativo ai parametri che regolano le altezze dei picchi dell'hrf, ma riguarda le ampiezze di tale curva (si veda per maggiori dettagli il sottoparagrafo 5.4.2).

Metodo 12: hrf costruita con lo spettro di potenza

Le varianti di questo metodo rispetto al 11 sono nella prima fase. Il tipo di selezione non viene più calcolata attraverso il GLM e quindi in modo dipendente da una specifica hrf, ma attraverso lo spettro di potenza delle frequenze presenti nei dati fMRI grezzi (si veda il sottoparagrafo 5.2.1). Come ci si può aspettare tale elaborazione è computazionalmente meno onerosa del GLM. Un'ulteriore differenza rispetto al metodo 11 sta nella terza fase ove nel caso in cui i vincoli applicati risultino troppo selettivi e quindi non vi siano voxel in uscita da tale fase, verranno considerate delle soglie meno rigide relative ai soli coefficienti di variazione; tutti i dettagli sono riportati nel sottoparagrafo 5.4.2.



5.2 Selezione dei voxel attivi

La selezione dei voxel attivi è la prima fase comune ai quattro metodi che svilupperemo nel corso del capitolo, essa si articola in due passi: la *selezione* e il *raggruppamento*.

5.2.1 Selezione

Per *selezione* si intende la rilevazione di un gruppo di voxel attivi dello specifico soggetto, tale insieme viene creato attraverso il metodo del GLM visto nel paragrafo 4.1 o attraverso lo spettro di potenza; analizziamo di seguito questo tipo di selezione.

Nello sviluppo del metodo dello spettro di potenza si è fatta l'ipotesi che tutti i voxel attivi presentassero un picco di potenza ad una frequenza fondamentale, quella del protocollo sperimentale. Infatti, siccome i soggetti sono stati stimolati ad una frequenza di 3 Hz, ossia con periodo pari a 1/3 di secondo in tale intervallo temporale la variazione della risposta emodinamica è trascurabile quindi possiamo considerarla costante per la maggior parte dell'event che nel nostro caso è di 26 s o 37 s. Da tutto ciò si deduce che lo spettro di potenza del segnale BOLD presenterà un picco alla frequenza del protocollo sperimentale nei voxel attivi. Nel protocollo in cui si utilizza un $TR=2.6$ s si cerca la frequenza $0.0192=1/52$ dove 52 s è il periodo di event-rest; nel protocollo con $TR=3.7$ s la frequenza cercata è $0.0135=1/74$ dove 74 s è il periodo event-rest.

In Cod.5.1 è riportato il codice che implementa tale spettro. Dalla riga 1 alla riga 16 si inizializzano tutti quei vettori e variabili che serviranno poi per la costruzione dello spettro; in particolare le frequenze minima e massima rilevabili in questo contesto sono rispettivamente $1/T$, dove $T=nT*TR$ è l'intera finestra di osservazione in secondi, e $(F_s-F_s/N)*1/T$, dove $nT=110$ volumi per i soggetti con $TR=2.6$ s e $nT=100$ per gli altri. Per poter rilevare lo spettro in modo corretto i campioni della finestra di osservazione sono stati posti pari a 100 in entrambi i casi. Dalla riga 17 alla 45 vengono considerati tutti i voxel all'interno della maschera, analizziamo la procedura per un singolo voxel. Per prima cosa ai dati grezzi relativi agli nT volumi viene sottratta la propria media (riga 23).

Successivamente si stima lo spettro di potenza adottando il *metodo diretto* in cui la densità spettrale di potenza $P(\omega)$ di un segnale discreto $x(n)$ con $n=0 \dots nT-1$ si ricava dalla relazione:

$$P(\omega) = \frac{1}{nT} |X(\omega)|^2 \quad (5.1)$$

dove $X(\omega)$ è la trasformata di Fourier discreta di $x(n)$ (righe 25-30). Siccome lo spettro di potenza è costituito da due parti simmetriche viene considerata solo la prima metà dello spettro (righe 31-32).

Nei casi considerati la frequenza cercata si trova sempre nella posizione 6 del vettore dello spettro quindi vengono scelti quei voxel che hanno la massima potenza nella posizione 6, ossia alla frequenza desiderata (riga 33). Successivamente dopo aver ordinato dalla potenza massima alla potenza minima il vettore dello spettro, chiamato s_ord , si scelgono quei voxel che soddisfano alla relazione:

$$\frac{s_ord(1)}{s_ord(1)} - \frac{s_ord(2)}{s_ord(1)} > 0.7 \quad (5.2)$$

ossia si scelgono quei voxel che nella frequenza desiderata hanno potenza superiore al 70% rispetto a tutte le altre considerate separatamente (riga 35). I voxel così selezionati sono quelli che poi verranno elaborati o attraverso il raggruppamento o direttamente attraverso i minimi quadrati non lineari nella seconda fase.

```

1) n=1;
2) [nR nC nfette nT]=size(matrix); % nR=numero righe, nC=numero colonne,
nfette=numero fette, nT=numero di campioni per ogni voxel
3) ts=TR;% intervallo di campionamento
4) T=nT*ts;% intervallo di osservazione
5) if nT==100,
6)     t=(0:ts:ts*nT-ts)';
7) elseif nT==110,
8)     t=(ts*nv_e:ts:ts*nT-ts)';
9) end
10) N=length(t);
11) Fs=100;% T/ts=frequenza di campionamento
    % vettore frequenze
12) f_FT=(0:(Fs/N)*(1/T):(Fs-Fs/N)*(1/T));
13) a=1;
14) conta=1;
15) [out,pixdim,dtype] = readanalyze('mask.img');
16) mask=out;
17) for fetta=1:nfette
18)     for i=1:nR
19)         for j=1:nC
20)             if mask(i,j,fetta)==1
21)                 mtr_p=reshape(matrix(i,j,fetta,:),1,nT);
22)                 mtr_pm=mean(mtr_p);
23)                 mtr_pf=mtr_p-mtr_pm;
24)                 n=n+1;
25)                 if nT==100,
26)                     FTx=fft(mtr_pf,N);
27)                 elseif nT==110,
28)                     FTx=fft(mtr_pf(11:110),N);
29)                 end
                % lo spettro  $P(\omega)$  per il voxel é dato da
30)                 s=(abs(FTx).^2)/N;
31)                 f_FT=f_FT(1:N/2+1);
32)                 s=s(1:N/2+1);
33)                 if s(6)==max(s)
34)                     s_ord=sort(s,'descend');
35)                     if ((s_ord(1)/s_ord(1))-(s_ord(2)/s_ord(1)))>0.7
36)                         val(a)=(s_ord(1)/s_ord(1))-(s_ord(2)/s_ord(1));
37)                         v_dif(a,:)=[i j fetta];
38)                         a=a+1;
39)                     end
40)                 end
41)                 conta=conta+1
42)             end
43)         end
44)     end
45) end
46) val_ord=sort(val,'descend');
47) a=0;
48) for i=1:length(val_ord)
49)     a=find(val==val_ord(i));
50)     v_dif_ord(i,:)=v_dif(a,:);
51) end
52) selezione=v_dif_ord;

```

Cod. 5.1: Codice relativo alla selezione con lo spettro di potenza.

5.2.2 Raggruppamento

I voxel ricavati dalla *selezione* alle volte si presentano in forma sparsa altre volte in gruppi ben delimitati e precisi. Siccome i voxel di interesse sono quelli che evidenziano intere zone allora si cerca di eliminare attraverso il raggruppamento i voxel isolati che sono quindi attivi con meno probabilità.

In questa tesi si è sviluppato il raggruppamento in due modi. Il primo, *raggruppa_vox*, utilizza la function di Matlab® *imfilter* che implementa il filtraggio spaziale lineare, di seguito verrà spiegato il codice relativo riportato in Cod 5.2. La function *raggruppa_vox* ha come *input field* le 36 matrici v_a , una per ogni fetta cerebrale, in cui i voxel di *selezione* sono contrassegnati con 1 e 0 tutti gli altri. Si è scelto come maschera del filtro w una matrice 3x3 i cui elementi hanno tutti valore 1, riga 5. Si è applicato il filtraggio spaziale ad ogni fetta, su tutto il volume cerebrale, in questo modo si assegna un peso ad ogni voxel in base al numero di adiacenze che ha su quella fetta. Ad esempio se un voxel con valore 1 di v_a ha altri due voxel adiacenti di valore 1 tale voxel assumerà peso 3, il valore di se stesso sommato al valore dei voxel adiacenti (righe 4-8).

In questo modo si sono rilevate le adiacenze sulla singola fetta; alle volte però vi sono voxel che risultano isolati all'interno della singola fetta, mentre costituiscono un gruppo se si prendono in considerazione le fette adiacenti. Quindi per quei voxel che presentano un numero di adiacenze inferiore a 4 si applica il filtraggio spaziale alla somma della fetta di interesse di v_a con le fette adiacenti; infine si procede nuovamente alla stessa sogliatura su tale voxel (righe 19-44).

L'*output field*, v_{at} , è un insieme di 36 matrici in cui i voxel in gruppo sono posti a 1, gli altri a 0. Questo modo di operare risulta piuttosto selettivo: ad esempio nel caso in cui un gruppo sia costituito da una porzione di riga o colonna o diagonale nella stessa fetta o combinazione analoga in fette diverse, a seguito del filtraggio, anche con soglie meno selettive di 4 (ad esempio 1), si perdono i voxel estremi del gruppo.

Per non incorrere in tale inconveniente si è sviluppato un secondo sistema di raggruppamento, la function *raggruppa_vox1*, con lo stesso *input field* del precedente che dopo aver applicato *imfilter* esegue l'operazione morfologica di *dilation*, ossia dà a tutti i voxel del gruppo lo stesso peso, cioè un peso pari al numero massimo di adiacenze più uno. L'effetto indesiderato della function di Matlab® *imdilate* è quello di aggiungere come attivi voxel al contorno del gruppo. Questo

problema viene eliminato moltiplicando per v_a il risultato della *dilation* (si veda Cod.5.3).

La soglia 4 adottata in *raggruppa_vox* è una soglia molto selettiva, visto che si vogliono selezionare dei voxel attivi con significatività elevata che siano rappresentativi per il soggetto. Alcuni pazienti, in cui non vi è attivazione, cioè sono caratterizzati da soli voxel sparsi, l'uscita del *raggruppa_vox* o del *raggruppa_vox1* non selezionerà alcun voxel; in questo caso verrà elaborata dal passo successivo l'uscita di *selezione* e verrà lanciato un flag, *fl1*.

```

1) function [v_at]= raggruppa_vox(v_a,flag)
2) [nR nC nfette]=size(v_a);
   % cerchiamo quei voxel attivi che hanno attorno altri voxel attivi con un
   raggio di 1 voxel
3) ris=zeros(nR,nC,nfette);
4) for i=1:nfette,
5)   w=ones(3,3); % maschera del filtro oppure w=ones(5,5);
6)   g=imfilter(v_a(:,:,i),w); % fa il filtraggio spaziale lineare con una
   maschera di 3x3
7)   ris(:,:,i)=v_a(:,:,i).*g; % matrice filtrata
8) end
9) if flag=='3x3',
   % poniamo a 1 tutti i voxel che in ris sono > di 1 e < di 5, e controlliamo che i
   % voxel con ris = 1,2,3,4 abbiano nelle fette adiacenti altri voxel attivi prima di
   tenerli come attivi altrimenti li poniamo a 0
10)  v_at=zeros(nR,nC,nfette); % matrice dei voxel attivi
11)  w1=ones(3,3); % maschera del filtro 3x3 per i voxel delle fette vicine
12)  for f=1:nfette,
13)    if sum(sum(ris(:,:,f)))~=0,
14)      for r=1:nR,
15)        for c=1:nC,
16)          if ris(r,c,f)>4, %1
17)            v_at(r,c,f)=1;
18)          end
19)          if ris(r,c,f)==1 || ris(r,c,f)==2 || ris(r,c,f)==3 || ris(r,c,f)==4,
%ris(r,c,f)==1
20)            if f==1,
21)              somma=sum(v_a(:,:,f:f+1),3); % matrice somma
22)              g1=imfilter(somma,w1); % fa il filtraggio con maschera 3x3
23)              if g1(r,c)>4, %1
24)                v_at(r,c,f)=1;
25)              end
26)              clear somma g1
27)            end
28)            if f>=2 && f<nfette-1,
29)              somma=sum(v_a(:,:,f-1:f+1),3); % matrice somma
30)              g3=imfilter(somma,w1); % fa il filtraggio con maschera 3x3
31)              if g3(r,c)>4, %1,
32)                v_at(r,c,f)=1;
33)              end
34)              clear somma g3
35)            end
36)            if f==nfette,
37)              somma=sum(v_a(:,:,f-1:f),3); % matrice somma
38)              g5=imfilter(somma,w1); % fa il filtraggio con maschera 3x3
39)              if g5(r,c)>4, %1,
40)                v_at(r,c,f)=1;
41)              end
42)              clear somma g5
43)            end
44)          end
45)        end
46)      end
47)    end
48)  end
49) end

```

Cod.5.2: Codice relativo al raggruppamento: primo metodo.

```

1)function [v_at]= raggruppa_vox1(v_a,flag)
.....CODICE OMESSO.....
4) for i=1:nfette,
5)   w=ones(3,3); % maschera del filtro oppure w=ones(5,5);
6)   g=imfilter(v_a(:, :, i),w).*v_a(:, :, i); % fa il filtraggio spaziale lineare con una
    maschera di 3x3
7)   ris(:, :, i)=imdilate(g,w).*v_a(:, :, i); % matrice filtrata con filtraggio spaziale
    lineare+dilation (dà a tutto il gruppo il valore del voxel con massima adiacenza)
8) end
.....CODICE OMESSO.....
16)       if ris(r,c,f)>2, %1
17)         v_at(r,c,f)=1;
18)       end
19)       if ris(r,c,f)==1 || ris(r,c,f)==2, %ris(r,c,f)==1
20)         if f==1,
21)           somma=sum(v_a(:, :, f:f+1),3); % matrice somma
22)           g1=imfilter(somma,w1); % fa il filtraggio spaziale lineare con
una maschera di 3x3
22a)         g2=imdilate(g1,w1);
23)         if g2(r,c)>2, %1
24)           v_at(r,c,f)=1;
25)         end
26)         clear somma g1 g2
27)       end
28)       if f>=2 && f<nfette-1,
29)         somma=sum(v_a(:, :, f-1:f+1),3); % matrice somma
30)         g3=imfilter(somma,w1); % fa il filtraggio spaziale lineare con
una maschera di 3x3
30a)         g4=imdilate(g3,w1);
31)         if g4(r,c)>2, %1,
32)           v_at(r,c,f)=1;
33)         end
34)         clear somma g3 g4
35)       end
36)       if f==nfette,
37)         somma=sum(v_a(:, :, f-1:f),3); % matrice somma
38)         g5=imfilter(somma,w1); % fa il filtraggio spaziale lineare con
una maschera di 3x3
38a)         g6=imdilate(g5,w1);
39)         if g6(r,c)>2, %1,
40)           v_at(r,c,f)=1;
41)         end
42)         clear somma g5 g6
43)       end
44)     end
.....CODICE OMESSO.....

```

Cod.5.3: Codice relativo al raggruppamento: varianti introdotte dal secondo metodo.

5.3 Descrizione del funzionamento della function *lsqnonlin* di MATLAB®

La seconda fase dei quattro metodi è caratterizzata dall'utilizzo della function di Matlab® *lsqnonlin*. Questa funzione è un ottimo strumento per la soluzione di problemi che richiedono in particolare i minimi quadrati non lineari.

Sia $F(x)$ la funzione obiettivo da minimizzare e x il vettore dei parametri che vogliamo stimare, allora tale problema di ottimizzazione si può scrivere come:

$$\min_x \frac{1}{2} \|F(x)\|_2^2 = \frac{1}{2} \sum_i f_i(x)^2$$

$$\text{dove } F(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_N(x) \end{bmatrix} \quad (5.3)$$

$F(x)$ è nel nostro caso un vettore di N istanti temporali, funzione dei parametri. Per risolvere tale problema di minimo *lsqnonlin* usa vari metodi simili all'algoritmo di Gauss-Newton come il *Trust region method* che utilizza a sua volta il *Preconditioned Conjugate Gradients (PCG)*, oppure l'algoritmo di *Levenberg-Marquardt* che abbiamo utilizzato nell'applicazione della funzione *lsqnonlin*; per questo motivo ci soffermeremo ora su tale metodo.

Algoritmo di Levenberg-Marquardt

L'algoritmo di Levenberg-Marquardt permette di ottenere una soluzione numerica al problema di minimizzazione di una funzione, spesso non lineare e dipendente da più parametri. L'algoritmo fonde insieme il metodo di Gauss-Newton (GNA), con quello del *gradient descent*, in questo modo il risultato è più robusto e trova una soluzione anche se il punto di partenza era molto lontano dal minimo. Tuttavia se vi sono funzioni con andamento regolare può convergere in modo leggermente più lento rispetto al metodo GNA.

La sua principale applicazione è nella regressione attraverso i minimi quadrati. La funzione da minimizzare è data da:

$$S(p) = \sum_{i=1}^N [y_i - f(x_i, p)]^2$$

(5.4)

dove nel nostro caso \mathbf{p} è il vettore dei parametri del modello $f(x_i, \mathbf{p})$ e y_i sono i dati relativi ad un singolo voxel all'istante i .

La procedura dell'algoritmo è iterativa, parte da un vettore iniziale di parametri \mathbf{p}_0 , ad esempio un vettore di parametri standard è $\mathbf{p}_0 = [1, 1, \dots, 1]^T$; in alcuni casi può non esserci convergenza se il vettore di partenza non è abbastanza vicino alla soluzione.

In ogni iterazione si sostituisce il vettore dei parametri \mathbf{p} con una nuova stima $\mathbf{p} + \delta$; per determinare δ le funzioni $f(x_i, \mathbf{p} + \delta)$ vengono approssimate attraverso lo sviluppo in serie di Taylor del primo ordine:

$$f(x_i, \mathbf{p} + \delta) \approx f(x_i, \mathbf{p}) + \mathbf{J}_i \delta \quad (5.5)$$

dove $\mathbf{J}_i = \frac{\partial f(x_i, \mathbf{p})}{\partial \mathbf{p}}$ cioè il gradiente di f rispetto a \mathbf{p} .

Una volta raggiunto il minimo della somma dei quadrati (S) il gradiente di S rispetto a \mathbf{p} è 0. Nel nostro caso, visto il campionamento, difficilmente avremo il gradiente di S uguale a 0 piuttosto si noterà un cambio di segno del gradiente quando la curva di $S(\mathbf{p})$ passerà da decrescente a crescente o viceversa.

Se si deriva la 5.4 rispetto ai parametri \mathbf{p} arrestandosi alle derivate prime e si pone il risultato uguale a zero si ottiene:

$$(\mathbf{J}^T \mathbf{J}) \delta = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})] \quad (5.6)$$

dove \mathbf{J} è la matrice Jacobiana in cui l' i -esima riga è \mathbf{J}_i ; \mathbf{f} e \mathbf{y} sono i vettori con i componenti y_i e $f(x_i, \mathbf{p})$. La 5.6 è un insieme di equazioni lineari che può essere risolto in δ . Poichè in qualche caso l'inversione $\mathbf{J}^T \mathbf{J}$ creava dei problemi Levenberg sostituì la 5.6 con la seguente versione ridotta:

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \delta = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})] \quad (5.7)$$

dove \mathbf{I} è la matrice identità e λ è il fattore di riduzione non negativo, detto *damping factor*, che viene aggiornato ad ogni iterazione. Se la riduzione di S è veloce viene adottato un valore più piccolo di λ , quindi l'algoritmo si avvicina all'algoritmo di Gauss-Newton. Se un'iterazione è poco efficace, cioè il residuo non viene ridotto rispetto al precedente, si può aumentare λ in questo modo ci si avvicina all'algoritmo del *gradient descent*.

L'algoritmo di Levenberg ha lo svantaggio che se il valore del *damping factor* λ è grande, l'inversione di $\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}$ crea delle difficoltà. Marquardt ebbe l'intuizione di scalare ogni componente del gradiente secondo la curvatura ciò conferisce un

movimento maggiore lungo le direzioni dove il gradiente è più piccolo in modo tale da evitare una convergenza lenta. Quindi Marquardt sostituì la matrice identità \mathbf{I} , con la diagonale della matrice Hessiana, $\mathbf{J}^T \mathbf{J}$, cioè:

$$(\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})) \delta = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\mathbf{p})] \quad (5.8)$$

Un simile parametro λ è stato utilizzato anche nel metodo di regolarizzazione di Tikhonov nel campo dei problemi lineari.

A questo punto dopo un certo numero di iterazioni, o quando ci si è avvicinati sufficientemente al minimo, l'algoritmo termina e restituisce come soluzione finale il vettore dei parametri \mathbf{p} .

Per ottenere una soluzione finale ottima è cruciale la stima del parametro λ ; Marquardt dice di iniziare da un valore λ_0 e un fattore $\nu > 1$. Si pone allora come valore di partenza $\lambda = \lambda_0$ e si calcola la somma dei quadrati delle deviazioni $S(\mathbf{p})$. Dopo un'iterazione si sostituisce $\lambda = \lambda/\nu$ e si procede con una seconda iterazione, se entrambe le iterazioni danno un punto peggiore rispetto al punto di partenza, allora si pone $\lambda = \lambda \nu$ e nelle iterazioni successive si moltiplica per ν fino a raggiungere un punto migliore con $\lambda = \lambda \nu^k$ per un certo k .

Se dall'uso del *damping factor* λ/ν si ha una riduzione del residuo quadrato allora λ/ν viene preso come nuovo valore di λ e il processo continua. Se utilizzando λ/ν dà una somma più grande, cioè residuo peggiore, ma usando λ dà un residuo migliore allora λ rimane immutato e il nuovo ottimo è preso come il valore ottenuto con λ come fattore di riduzione.[14]

5.3.1 Applicazione di *lsqnonlin* ai quattro metodi sperimentati

La funzione *lsqnonlin* di Matlab[®] necessita di una function *objective* per processare i voxel, ossia una specie di funzione obbiettivo che ha come *input field* il vettore corrente dei parametri **p**, nel caso dei metodi 5, 6, 11 di dimensione 7, il vettore dei tempi in secondi, **temp**, della funzione con cui costruiremo il modello e il vettore dei dati **data** di cui si fa il fit. L'*output field* **vettOut** è del tipo : $y_i - f(x_i, \mathbf{p})$ e nel nostro codice è lo stesso per tutti i quattro metodi (Cod. 5.4, riga 10). **KWmodello** che corrisponde a $f(x_i, \mathbf{p})$ è il vettore del modello filtrato con la matrice **K** e pesato con la matrice **W** (riga 9). Il vettore **data** è il vettore dei dati filtrati e pesati allo stesso modo del modello e corrisponde a y_i .

```

1) function vettOut = objective2 (p, temp, data)

2) load dat_obj
    % La funzione obbiettivo dipende da nT, ossia dal numero di campioni per
    % ogni voxel da K e W
    % HRF Handwerker con i parametri p
3) TR=temp(2); % frequenza di campionamento
4) tf=temp(length(temp))+TR; % periodo active-rest in secondi
5) delta2=p(1)+p(2)+p(2)/2; % si è fissato per l'elaborazione di lsqnonlin
    delta2=delta1+tau1+tau1/2;
6) h_f=hrf_HW(TR,tf,p(1),p(2),p(3),delta2,p(4),p(5),p(6),1,'two_gamma_der');
7) C=p(7); % valore dello zero del modello
8) modello=ProtMod1(nT,h_f,C); % costruzione modello
9) KWmodello=W*modello-K*(K'*W*modello);
    % fit del modello con i dati
10) vettOut = (KWmodello - data);

```

Cod.5.4: Funzione obbiettivo per i metodi 5 e 6.

Di seguito analizzeremo la procedura del *lsqnonlin* per i metodi 5 e 6. In Cod.5.4 righe 5-6 viene costruita l'hrf_{fin} di Handwerker con 6 parametri in particolare nella riga 5 si è fissato il parametro delta2 in modo da evitare gli effetti indesiderati risultanti dalla somma della curva positiva e negativa. Nella riga 8 ci si costruisce il modello attraverso il solito protocollo event-rest visto nel capitolo 2, il parametro C viene sommato alla convoluzione tra stimolo e hrf e ha la funzione di far coincidere la *baseline* del modello con quella dei dati.

```

% Ordine dei parametri
% i=1: parametro delta1
% i=2: parametro tau1
% i=3: parametro a1
% i=4: parametro tau2
% i=5: parametro a2
% i=6: parametro delta
% i=7: parametro C
% i=8: parametro exitFlag
1) estimates = zeros(nR, nC, nfette, 8) ; %8 = numero di parametri + exitFlag
2) options
    =optimset('Display','off','LargeScale','on','LevenbergMarquardt','on','MaxFunEv
als',10e6,'MaxIter',1500,'TolFun',1e-6,'TolX',1e-6 );

3) p0=[2.6; 7.8; 20; 7; 1; 1; 656]; % parametri iniziali
4) lb=[0; 0; 0; p0(2)/3; 0; 0; 0]; % lower bound
5) ub=[10; 20; 200; 20; p0(3)/2; 5; 1500]; % upper bound
% LSQNONLIN
6) n_parametri=7;
7) residui=zeros(nR,nC,nfette,nT);
8) residNorm=zeros(nR,nC,nfette);
9) hrf_fin=zeros(nR,nC,nfette,length(t));
10) KWmod=zeros(nR,nC,nfette,nT);
11) CV_estimates=zeros(nR,nC,nfette,n_parametri);
12) estimates=zeros(nR,nC,nfette,n_parametri);
13) vettData1=zeros(nR,nC,nfette,nT);
14) par_hrf=zeros(sum(sum(sum(v_at))),3+n_parametri);
15) i=1;
16) for fetta=1:nfette,
17)     for r = 1 : nR,
18)         for c = 1 : nC,
19)             if sum(sum(v_at(:, :, fetta))) ~ =0,
20)                 if v_at(r,c,fetta) ==0,
21)                     p = [0.8; 7.8; 0; 12; 0; 1; 0];
22)                     exitFlag = 5 ;

```

Cod.5.5a: Codice relativo alla stima ai minimi quadrati non lineari per i metodi 5 e 6.

```

23)     else
24)         vettData2 = reshape(matrix(r, c,fetta, :), nT, 1);
25)         vettData=W*vettData2-K*(K'*W*vettData2); % vett colonna (nTx1)
26)         vettData1(r,c,fetta,:)=vettData;
27)         [p, resNorm, residual, exitFlag, output,lambda,jacobian] =
lsqnonlin(@objective2,      p0, lb, ub, options, t, vettData) ;
28)         residui(r,c,fetta,:)=residual;
29)         residNorm(r,c,fetta)=resNorm;
30)         estimates(r, c, fetta,1) = p(1)      ; % delta1
31)         estimates(r, c, fetta,2) = p(2)      ; % tau1
32)         estimates(r, c, fetta,3) = p(3)      ; % a1
33)         estimates(r, c, fetta,4) = p(4)      ; % tau2
34)         estimates(r, c, fetta,5) = p(5)      ; % a2
35)         estimates(r, c, fetta,6) = p(6)      ; % delta
36)         estimates(r, c, fetta,7) = p(7)      ; % C
37)         estimates(r, c, fetta,8) = exitFlag ; % condizione di uscita
% matrice con i parametri dell'hrf + la posizione del voxel
38)         par_hrf(i,1)=r;
39)         par_hrf(i,2)=c;
40)         par_hrf(i,3)=fetta;
41)         par_hrf(i,4)=p(1);
42)         par_hrf(i,5)=p(2);
43)         par_hrf(i,6)=p(3);
44)         par_hrf(i,7)=p(4);
45)         par_hrf(i,8)=p(5);
46)         par_hrf(i,9)=p(6);
47)         par_hrf(i,10)=p(7);
48)         i=i+1;
% CALCOLO PRECISIONI - pesaggio relativo
49)         sigma2=resNorm/(nT-n_parametri);
50)         format short g
51)         CovErrStimaParametri=sigma2*inv(full(jacobian)'*W*full(jacobian));
52)         SD_p_stimati=sqrt(diag(CovErrStimaParametri));

53)         CV_p_stimati=100*abs(SD_p_stimati./p);
54)         CV_estimates(r,c,fetta,1)=CV_p_stimati(1);
55)         CV_estimates(r,c,fetta,2)=CV_p_stimati(2);
56)         CV_estimates(r,c,fetta,3)=CV_p_stimati(3);
57)         CV_estimates(r,c,fetta,4)=CV_p_stimati(4);
58)         CV_estimates(r,c,fetta,5)=CV_p_stimati(5);
59)         CV_estimates(r,c,fetta,6)=CV_p_stimati(6);
60)         CV_estimates(r,c,fetta,7)=CV_p_stimati(7);
61)         disp('=====')
62)         disp(['fetta= ',num2str(fetta),' riga = ',num2str(r),' colonna=
',num2str(c)])
63)         disp(' ')
64)         disp(['exitflag=',num2str(exitFlag)])
65)         disp(['delta1=',num2str(p(1)),' tau1=',num2str(p(2)),'
a1=',num2str(p(3)),' tau2=',num2str(p(4)),' a2=',num2str(p(5)),'
delta=',num2str(p(6)),' C=',num2str(p(7))])
66)         disp(['CV_delta1=',num2str(CV_p_stimati(1)),'
CV_tau1=',num2str(CV_p_stimati(2)),' CV_a1=',num2str(CV_p_stimati(3)),'
CV_tau2=',num2str(CV_p_stimati(4)),' CV_a2=',num2str(CV_p_stimati(5)),'
CV_delta=',num2str(CV_p_stimati(6)),' CV_C=',num2str(CV_p_stimati(7))])

```

Fig.5.5b: Codice relativo alla stima ai minimi quadrati non lineari per i metodi 5 e 6.

```

% Ricalcolo il modello con i p ottenuti e calcolo l'ampiezza
% del modello
% HRF Handwerker con i parametri appena stimati
67) delta2=p(1)+p(2)+p(2)/2;
68)
h_f=hrf_HW(TR,tf,p(1),p(2),p(3),delta2,p(4),p(5),p(6),1,'two_gamma_der');
69) hrf_fin(r,c,fetta,:)=h_f;
% creazione del modello pesato
70) modello=ProtMod1(nT,h_f,p(7));
71) KWmodello=W*modello-K*(K'*W*modello);
72) KWmod(r,c,fetta,:)=KWmodello;
73) end
74) end
75) end
76) end
77) end

```

Fig.5.5c: Codice relativo alla stima ai minimi quadrati non lineari per i metodi 5 e 6.

In Cod.5.5a,b,c è riportato il codice relativo all'applicazione di *lsqnonlin*; in particolare alla riga 2 attraverso la function di Matlab® *optimset* si scelgono gli algoritmi che verranno poi effettivamente utilizzati da *lsqnonlin*; nel nostro caso si è utilizzato l'algoritmo di Levenberg-Marquardt e tutte le function relative all'opzione *Large Scale*. Per quanto riguarda i limiti: *MaxFunEvals* è il massimo numero di funzioni valutate, *MaxIter* il numero massimo di iterazioni, *TolFun* e *TolX* sono le tolleranze di termine del valore della funzione e dei parametri rispettivamente. Nelle righe 3-5 vengono fissati i valori iniziali dei parametri e i range di validità relativi. Dalla riga 6 alla 15 sono stati inizializzati tutti i vettori, variabili e matrici che in seguito saranno usati. Solo i voxel selezionati nella fase 1 che sono stati posti a 1 nelle 36 matrici di *v_at* vengono ora processati in questa fase dalla riga 23 alla 77. Nelle righe 24-25 ci si costruisce il vettore **vettData**, ossia il vettore **data** passato come ingresso alla function *objective2*, a *lsqnonlin* viene passato come ingresso anche il vettore **t** che corrisponde al vettore **temp input field** di *objective2*. I dati in uscita dall'elaborazione saranno: il vettore dei parametri **p**, che servirà ad ottenere l'hrf ottima, *resNorm*, corrispondente al WRSS, che è utilizzato per la stima vista nella 4.12 di σ^2 (riga 49), infine la matrice **J**. Tale matrice serve per calcolare, secondo la 4.14, la matrice di covarianza dell'errore di stima dei parametri (riga 51), la deviazione standard dei parametri stimati (riga 52) e quindi i coefficienti di variazione degli stessi (riga 53) come dalla 4.16.

I metodi 11 e 12 hanno la stessa struttura base dei metodi 5 e 6; perciò si è preferito riportare solo le parti di codice che sono state modificate mantenendo sempre la

```

1) function vettOut = objective3 (p, temp, data)

2) load dat_obj
% La funzione obbiettivo dipende da nT, ossia dal numero di campioni per
% ogni voxel da K e W
% HRF Handwerker con i parametri p
3) TR=temp(2); % frequenza di campionamento
4) tf=temp(length(temp))+TR; % periodo active-rest in secondi
5) delta2=p(1)+p(2)+p(2)/2; % si è fissato per l'elaborazione di lsqnonlin
    delta2=delta1+tau1+tau1/2;
6) tau2=p(2)/2+p(4);
7) a1=2*p(5)+p(3);
8) h_f=hrf_HW(TR,tf,p(1),p(2),a1,delta2,tau2,p(5),p(6),1,'two_gamma_der');
9) C=p(7); % valore dello zero del modello
10) modello=ProtMod1(nT,h_f,C); % costruzione modello
11) KWmodello=W*modello-K*(K'*W*modello);
    % fit del modello con i dati
12) vettOut = (KWmodello - data);

```

Cod.5.6: Funzione objective comune ai metodi 11 e 12.

stessa numerazione delle righe del codice dei metodi 5 e 6. Le righe aggiunte portano lo stesso numero dell'ultima riga di codice che precede il loro inserimento e al numero si è affiancata una lettera.

La funzione *objective* dei metodi 11 e 12 (Cod.5.6) coincide con quella dei metodi precedenti salvo per le righe 6-8 in cui per eliminare ulteriori distorsioni dell'hrf si è fissato il limite inferiore di τ_2 provvedendo poi a incrementarlo della quantità positiva $\Delta\tau_2$ come segue:

$$\tau_2 = \frac{\tau_1}{2} + \Delta\tau_2 \quad (5.9)$$

analogamente a_1 è stata posta pari a:

$$a_1 = 2*a_2 + \Delta a_1. \quad (5.10)$$

Tali parametri vengono poi sostituiti nella riga 8 per la determinazione dell'hrf. Questa modifica viene ripetuta anche all'interno della procedura del *lsqnonlin* dei metodi 11 e 12 (righe 67a,67b,68 di Cod.5.7). Ne consegue che, essendo diversi i due parametri, sono stati modificati anche i range di validità di questi stessi (righe 4 e 5).

Per quanto riguarda il metodo 6, la procedura per la stima dei parametri δ_1 , τ_1 , Δa_1 , $\Delta\tau_2$, a_2 , δ e C è la stessa del metodo 5. In questo caso però vi è la stima dell'ulteriore parametro m_2 , ossia il coefficiente angolare del secondo asintoto dell'iperbole, riportata nel capitolo 3. Per valutare m_2 , poiché lo spostamento laterale è determinato principalmente dai parametri δ_1 , τ_1 , τ_2 e m_2 stesso si sono dovuti mantenere costanti i

```

% Ordine dei parametri:
%   i=1: parametro delta1
%   i=2: parametro tau1
%   i=3: parametro da1
%   i=4: parametro dtau2
%   i=5: parametro a2
%   i=6: parametro delta
%   i=7: parametro C
%   i=8: parametro exitFlag
4) lb=[0; 0; 0; 0; 0; 0; 0; 0]; % lower bound
5) ub=[10; 20; 200; 20; 100; 5; 1500]; % upper bound
% LSQNONLIN
.....CODICE OMESSO.....
27) [p, resNorm, residual, exitFlag, output,lambda,jacobian] =
lsqnonlin(@objective3, p0, lb, ub, options, t, vettData) ;
.....CODICE
OMESSO.....
32) estimates(r, c, fetta,3) = p(3) ; % da1
33) estimates(r, c, fetta,4) = p(4) ; % dtau2
.....CODICE OMESSO.....
65) disp(['delta1=',num2str(p(1)),' tau1=',num2str(p(2)),'
a1=',num2str(2*p(5)+p(3)),' tau2=',num2str(p(2)/2+p(4)),' a2=',num2str(p(5)),'
delta=',num2str(p(6)),' C=',num2str(p(7))])
.....CODICE OMESSO.....
% Ricalcolo il modello con i p ottenuti e calcolo l'ampiezza del modello
% HRF Handwerker con i parametri appena stimati

67) delta2=p(1)+p(2)+p(2)/2;% si è fissato per l'elaborazione di lsqnonlin
delta2=delta1+tau1+tau1/2;
67a) tau2=p(2)/2+p(4);
67b) a1=2*p(5)+p(3);
68) h_f=hrf_HW(TR,tf,p(1),p(2),a1,delta2,tau2,p(5),p(6),1,'two_gamma_der');
.....CODICE OMESSO.....

```

Cod.5.7: Codice relativo alla stima ai minimi quadrati non lineari per il metodo 11.

```

1) function vettOut = objective12(p1, temp, data)
% funzione obbiettivo per l'iperbole
2) load dat_obj % carica K, W, nT
3) load dat_p % carica il vettore dei parametri p
% La funzione obbiettivo dipende da nT, ossia dal numero di campioni per
% ogni voxel da K e W
% HRF Handwerker con i parametri p
4) TR=temp(2); % frequenza di campionamento
5) tf=temp(length(temp))+TR; % periodo active-rest in secondi
6) delta2=p(1)+p(2)+p(2)/2; % si è fissato per l'elaborazione di lsqnonlin
delta2=delta1+tau1+tau1/2;
7) tau2=p(2)/2+p(4);
8) a1=2*p(5)+p(3);
9) h_f=hrf_HW2(TR,tf,p(1),p(2),a1,delta2,tau2,p(5),p(6),p1,'two_gamma_der');
10) C=p(7); % valore dello zero del modello
11) modello=ProtMod1(nT,h_f,C); % costruzione modello
12) KWmodello=modello-K*(K'*modello);
% fit del modello con i dati
13) vettOut = (KWmodello - data);

```

Cod.5.8: Funzione objective relativa alla stima di m_2 nel metodo 12.

primi tre facendo variare separatamente m_2 ; in questo modo, quando serve, si crea un'ulteriore dispersione. In Cod.5.9 sono riportate le sole parti modificate dell'implementazione relativa alla procedura *lsqnonlin*, mentre in Cod.5.8 è riportato il codice della function *objective* che riguarda l'elaborazione del solo parametro m_2 .

```
% Ordine dei parametri:
%   i=1: parametro delta1
%   i=2: parametro tau1
%   i=3: parametro a1
%   i=4: parametro tau2
%   i=5: parametro a2
%   i=6: parametro delta
%   i=7: parametro C
%   i=8: parametro m2
%   i=9: parametro exitFlag
.....CODICE OMESSO.....
5a) p01=0.25;
5b) lb1=0.25;
5c) ub1=1.5;
.....CODICE OMESSO.....
60a) [p1, resNorm1, residual1, exitFlag1, output1,lambda1,jacobian1] =
lsqnonlin(@objective12, p01, lb1, ub1, options, t, vettData) ;
60b) estimates(r, c, fetta,8) = p1 ; % m2
60c) par_hrf(i,11)=p1;
60d) i=i+1;
60e) sigma21=resNorm1/(nT-1);
60f) format short g
60g) CovErrStimaParametri1=sigma21*inv(full(jacobian1)*full(jacobian1));
60h) SD_p_stimati1=sqrt(diag(CovErrStimaParametri1));
60i) CV_estimates(r,c,fetta,8)=100*abs(SD_p_stimati1./p1);
.....CODICE OMESSO.....
65) disp(['delta1=',num2str(p(1)), ' tau1=',num2str(p(2)), '
a1=',num2str(2*p(5)+p(3)), ' tau2=',num2str(p(2)/2+p(4)), '
a2=',num2str(p(5)), ' delta=',num2str(p(6)), ' C=',num2str(p(7)), '
m2=',num2str(p1)])
66) disp(['CV_delta1=',num2str(CV_p_stimati(1)), '
CV_tau1=',num2str(CV_p_stimati(2)), ' CV_a1=',num2str(CV_p_stimati(3)), '
CV_tau2=',num2str(CV_p_stimati(4)), ' CV_a2=',num2str(CV_p_stimati(5)), '
CV_delta=',num2str(CV_p_stimati(6)), ' CV_C=',num2str(CV_p_stimati(7)), '
CV_m2=',num2str(CV_estimates(r,c,fetta,8))])
% Ricalcolo il modello con i p ottenuti e calcolo l'ampiezza del modello
% HRF Handwerker con i parametri appena stimati
67) delta2=p(1)+p(2)+p(2)/2;
67a) tau2=p(2)/2+p(4);
67b) a1=2*p(5)+p(3);
68) h_f=hrf_HW2(TR,tf,p(1),p(2),a1,delta2,tau2,p(5),p(6),p1,'two_gamma_der');
.....CODICE OMESSO.....
```

Cod.5.9: Codice relativo alla stima ai minimi quadrati non lineari per il metodo 12.

5.4 Vincoli e procedure per la determinazione dell'hrf specifica del soggetto

La terza fase è costituita dai vincoli per la selezione dei voxel ottimi, ossia quei voxel su cui si procederà poi a fare la media pesata o non pesata per determinare l'hrf specifica del soggetto. Ora vedremo in dettaglio i vincoli utilizzati nei metodi 5 e 6, vincoli sulle altezze, e quelli utilizzati nei metodi 11 e 12, vincoli sulle ampiezze.

5.4.1 Vincoli sulle altezze

Inizialmente si è scelto di selezionare i voxel ottimi in base all'altezza del

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SELEZIONE DELL'HRF OTTIMA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1) j=1;
2) for i=1:size(par_hrf,1),
3)   r=par_hrf(i,1);
4)   c=par_hrf(i,2);
5)   f=par_hrf(i,3);
6)       if (par_hrf(i,6)<par_hrf(i,8)) || isinf(CV_estimates(r,c,f,1)) ||
CV_estimates(r,c,f,3)>9000 || CV_estimates(r,c,f,5)>9000, % elimino i voxel con
(a1<a2) o CV=inf o con CV_a1>9000 o CV_a2>9000
7)       par_hrf(i,:)=zeros(1,3+n_parametri);
8)   else
9)       par_hrf1(j,:)=par_hrf(i,:); % serve per la media dei parametri dell'hrf
10)      CV_hrf1(j,:)=reshape(CV_estimates(r,c,f,:),1,n_parametri);
11)      j=j+1;
12)   end
13) end

```

Cod. 5.10: Codice relativo ai vincoli utilizzati nei metodi 5 e 6 per la costruzione dell'hrf ottima.

picco positivo a_1 e del picco negativo a_2 dell'hrf secondo il principio che i migliori voxel attivi dovevano avere come risposta allo stimolo una maggiore differenza tra la fase di active e quella di rest rispetto a voxel non attivi o meno attivi. Inoltre, tenendo conto che l'errore rappresenta una componente considerevole del dato si è considerato il CV di tali parametri come indice della validità della stima.

In particolare si è scelto, in modo empirico, di tenere soltanto quei voxel che presentavano $a_1 > a_2$, in quanto l'hrf risultante con $a_1 < a_2$ non era una risposta emodinamica fisiologicamente plausibile. Inoltre si è deciso di vincolare i CV dei

parametri quale indice della bontà delle stime dei parametri stessi. In particolare si sono eliminati tutti quei voxel che avevano uno o più CV tendenti ad infinito e quelli che presentavano il coefficiente di variazione di a_1 o di a_2 maggiore di 9000, si è scelta tale soglia analizzando i risultati della seconda fase per alcuni soggetti. Il codice Matlab[®] che implementa questi vincoli è riportato in Cod.5.10.

Una volta selezionati i voxel ottimi, l'esigenza è quella di trovare una sola curva della risposta emodinamica rappresentativa delle caratteristiche del soggetto da utilizzare nell'elaborazione finale con il metodo GLM. È necessario inoltre che questa curva riporti le peculiarità comuni ai singoli voxel selezionati, perciò si è deciso di ottenerla attraverso la media. L'hrf media può essere di due tipi; media delle hrf negli istanti temporali, oppure un'hrf costituita dalla media dei parametri. Al confronto con i dati, i risultati migliori si sono ottenuti con quest'ultima che quindi è stata utilizzata nel seguito come riportato in Cod.5.11 (righe 1-3).

Un'ulteriore variante può essere una media pesata sui coefficienti di variazione dei singoli parametri in modo tale che i parametri con CV più alti, cioè parametri poco rappresentativi, pesino di meno nel calcolo della media. In Cod.5.11 (righe 4-42) è stato riportato il codice relativo a tale media, la quale è stata utilizzata solo nel metodo 6 in quanto si è visto che risultava troppo selettiva. In particolare, anche se è estremamente improbabile che il valore di un CV risulti uguale a zero, per evitare che il valore finale dei parametri dopo la pesatura sia Not a Number (NaN) si controlla questa evenienza (righe 16-21). Dalla riga 22 alla 33 e nella riga 35 viene calcolata la media pesata di ciascun parametro secondo la seguente formula:

$$p_{\text{pesato}}(i) = \frac{\frac{1}{CV(p_1(i))} * p_1(i) + \dots + \frac{1}{CV(p_n(i))} * p_n(i)}{\frac{1}{CV(p_1(i))} + \dots + \frac{1}{CV(p_n(i))}}$$

(5.11)

dove $p_{\text{pesato}}(i)$ è la media pesata dell' i -esimo parametro, $p_1(i), \dots, p_n(i)$ sono i valori dell' i -esimo parametro nelle n hrf dei voxel selezionati e $CV(p_1(i)), \dots, CV(p_n(i))$ sono i relativi coefficienti di variazione.


```

    % faccio la media dei parametri delle hrf rimaste
1) par_hrf_medi=mean(par_hrf1,1);
    %hrf_medie=mean(hrf_fin1,1);
2) delt2=par_hrf_medi(4)+par_hrf_medi(5)+(par_hrf_medi(5)/2);
3)
hrf_media=hrf_HW(TR,tf,par_hrf_medi(4),par_hrf_medi(5),par_hrf_medi(6),delt2,par_hrf_medi(7),
    par_hrf_medi(8),par_hrf_medi(9),1,'two_gamma_der');
    % faccio la media pesata sui CV dei parametri delle hrf rimaste
4) s_delta1=0; % inizializzo tutte le variabili somma
5) s_cv_d1=0;
6) s_tau1=0;
7) s_cv_tau1=0;
8) s_a1=0;
9) s_cv_a1=0;
10) s_tau2=0;
11) s_cv_tau2=0;
12) s_a2=0;
13) s_cv_a2=0;
14) s_delta=0;
15) s_cv_d=0;
16) for i=1:size(par_hrf1,1),
17)     for j=1:n_parametri-1,
18)         if CV_hrf1(i,j)==0,
19)             CV_hrf1(i,j)=1;
20)         end
21)     end
22)     s_delta1=s_delta1+(1/CV_hrf1(i,1))*par_hrf1(i,4); % somma pesata dei
delta1
23)     s_cv_d1=s_cv_d1+(1/CV_hrf1(i,1)); % somma dei pesi dei delta1
24)     s_tau1=s_tau1+(1/CV_hrf1(i,2))*par_hrf1(i,5); % somma pesata dei tau1
25)     s_cv_tau1=s_cv_tau1+(1/CV_hrf1(i,2)); % somma dei pesi dei tau1
26)     s_a1=s_a1+(1/CV_hrf1(i,3))*par_hrf1(i,6); % somma pesata degli a1
27)     s_cv_a1=s_cv_a1+(1/CV_hrf1(i,3)); % somma dei pesi degli a1
28)     s_tau2=s_tau2+(1/CV_hrf1(i,4))*par_hrf1(i,7); % somma pesata dei tau2
29)     s_cv_tau2=s_cv_tau2+(1/CV_hrf1(i,4)); % somma dei pesi dei tau2
30)     s_a2=s_a2+(1/CV_hrf1(i,5))*par_hrf1(i,8); % somma pesata degli a2
31)     s_cv_a2=s_cv_a2+(1/CV_hrf1(i,5)); % somma dei pesi degli a2
32)     s_delta=s_delta+(1/CV_hrf1(i,6))*par_hrf1(i,9); % somma pesata dei delta
33)     s_cv_d=s_cv_d+(1/CV_hrf1(i,6)); % somma dei pesi dei delta
34) end
35) par_hrf_m_pes=[s_delta1/s_cv_d1 s_tau1/s_cv_tau1 s_a1/s_cv_a1
s_tau2/s_cv_tau2 s_a2/s_cv_a2 s_delta/s_cv_d]; % media pesata dei
parametri
36) delt2=par_hrf_m_pes(1)+par_hrf_m_pes(2)+(par_hrf_m_pes(2)/2);
    % hrf ottenuta con la media pesata dei parametri
37)
hrf_m_pes=hrf_HW(TR,tf,par_hrf_m_pes(1),par_hrf_m_pes(2),par_hrf_m_pes(3),delt2,
    par_hrf_m_pes(4),par_hrf_m_pes(5),par_hrf_m_pes(6),1,'two_gamma_der'
);
38) h=hrf_m_pes;

```

Cod. 5.11: Codice relativo alla costruzione dell'hrf pesata e non pesata.

5.4.2 Vincoli sulle ampiezze

Si è notato in seguito che i parametri a_1 e a_2 , relativi all'altezza dei picchi dell'hrf, vengono ulteriormente adattati ai dati con la successiva elaborazione del GLM attraverso i parametri β . Le ampiezze che dipendono invece direttamente dai parametri τ_1 e τ_2 non vengono considerati dal GLM, ma soltanto dal lsqnonlin. Perciò nei metodi 11 e 12 si è preferito procedere ad una selezione basata su questi parametri. In particolare dopo l'osservazione su diversi soggetti affetti da ictus si sono scelti i vincoli su τ_1 e τ_2 che sono stati riportati in Cod.5.12 (riga 6) e sono i seguenti:

$$0.2 < \frac{\tau_1}{\tau_2} < 0.8; \quad 13 < \tau_1 + \tau_2 < 30; \quad \tau_1 > 3; \quad \text{con} \quad CV_{\tau_1},$$

$$CV_{\tau_2} < 1000.$$

Inoltre non vengono considerati nella scelta anche quei voxel che presentano almeno un CV tendente ad *inf*.

```

1) j=1;
2) for i=1:size(par_hrf,1),
3)   r=par_hrf(i,1);
4)   c=par_hrf(i,2);
5)   f=par_hrf(i,3);
6)   if isinf(CV_estimates(r,c,f,1)) || CV_estimates(r,c,f,2)>1000 ||
       CV_estimates(r,c,f,4)>1000 || par_hrf(i,5)/((par_hrf(i,5)/2+par_hrf(i,7)))<0.2
       || par_hrf(i,5)/((par_hrf(i,5)/2+par_hrf(i,7)))>0.8 || par_hrf(i,5)<3 ||
       (par_hrf(i,5)+(par_hrf(i,5)/2+par_hrf(i,7)))>30 ||
       (par_hrf(i,5)+(par_hrf(i,5)/2+par_hrf(i,7)))<13, %CV=inf o con CV_tau1>1000
       o CV_tau2>1000 o con tau1/tau2<0.2 o tau1/tau2>0.8 o con o con tau1<3 o con
       tau1+tau2<13 o tau1+tau2>30
7)     par_hrf(i,:)=zeros(1,3+n_parametri);
8)   else
9)     par_hrf1(j,:)=par_hrf(i,:); % serve per la media dei parametri dell'hrf
10)    j=j+1;
11) end
12) end

```

Cod.5.12: Codice di scelta dei voxel ottimi per la media finale nei metodi 11 e 12.

In alcuni soggetti può capitare che la curva caratteristica sia molto differente rispetto ai vincoli posti sui parametri, in quanto questi ultimi erano stati studiati appositamente per soggetti affetti da stroke. In altre parole può essere che nessun voxel passi i vincoli in tal caso il metodo 12 lancia un flag, (*fl3*) e procede con una sogliatura meno rigida, ossia si sono tolti i vincoli sui τ mentre rimangono quelli sui CV (Cod.5.13).

In tutti i metodi se non c'è almeno un voxel che supera i vincoli della fase tre viene lanciato un flag (*fl2*) e questa volta verranno elaborati nella seconda fase tutti i voxel di *selezione*.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SELEZIONE DELL'HRF OTTIMA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%n_parametri=7;
1) j=1;
2) for i=1:size(par_hrf,1),
3)   r=par_hrf(i,1);
4)   c=par_hrf(i,2);
5)   f=par_hrf(i,3);
6)   if isinf(CV_estimates(r,c,f,1)) || CV_estimates(r,c,f,2)>1000 ||
CV_estimates(r,c,f,4)>1000,
7)     par_hrf(i,:)=zeros(1,3+n_parametri);
8)   else
9)     par_hrf1(j,:)=par_hrf(i,:); % serve per la media dei parametri dell'hrf
10)    j=j+1;
11) end
12) end

```

Cod.5.13: Codice di scelta dei voxel ottimi per la media finale nel metodo 12.

Capitolo 6

Risultati

Per decidere quale dei metodi trattati nei capitoli precedenti dà i risultati migliori dobbiamo stabilire un criterio di confronto che non penalizzi un metodo rispetto ad un altro e che si basi su indici ricavati matematicamente. Quindi si è scelto di calcolare l'indice di Akaike (AIC) come base del confronto. È logico attendersi che tale indice sia diverso da voxel a voxel per cui si è scelto di utilizzare la media degli indici per i voxel attivi, inoltre per rendere tale media indicativa di tutti i metodi deve essere applicata ai soli voxel comuni. Ci si attende che il metodo migliore risulti essere quello con AIC medio più basso secondo il Criterio di Parsimonia. Nell'analisi dei risultati si è deciso di non tenere in considerazione il paziente numero 10, in quanto durante la fase di acquisizione si è mosso oltre la correzione possibile, quindi il nostro data set si riduce a 19 soggetti.

6.1 Primo confronto

Per attuare un primo confronto (codice 6.1a,b,c) si è scelta l'hrf migliore tra le quattro hrf standard del Brain Voyager (metodo BVXtool), e la si è confrontata con tutti gli altri metodi, ossia quelli visti nel capitolo 4 ad hrf fissa e quelli del capitolo 5 ad hrf specifica per il soggetto (tabella 6.1). Dagli indici di AIC medi si nota che il metodo BVXtool ha la meglio in 4 casi su 18; per i soggetti 7 e 9 non si è potuto calcolare tale indice per mancanza di voxel comuni. In queste considerazioni si deve tenere conto anche che l'indice di AIC nel vecchio metodo è stato calcolato in modo diverso, ossia attraverso la formula:

$$AIC = N * \log(WRSS) + 2 * M \quad [22]$$

(6.1)

dove N è il numero di volumi e M il numero dei parametri, con tale formula si ottengono valori leggermente più piccoli rispetto alla formula 4.13 utilizzata in tutti gli altri metodi. Inoltre si deve anche tener presente che la hrf di Boynton, implementata in Brain Voyager, è migliore in 2 dei 4 casi favorevoli e ricalca quasi esattamente la single gamma HW (figura 3.5). I risultati ottenuti con tale hrf e AIC nuovo dell'eq. 4.13 sono quelli ottenuti dal metodo 7. Per questi motivi si è deciso di confrontare tra loro solo i metodi che utilizzano le hrf di Handwerker. Inoltre nel confronto finale a causa degli scarsi risultati sono stati eliminati anche i metodi 5 e 6 relativi alle hrf con media dei parametri e hrf con media pesata dei parametri.

Tab.6.1: Primo confronto.

Soggetti	n. voxel comuni		1-4)HRF BV migliore	5)HRF media dei parametri	6)HRF media pesata dei parametri	7)HRF B HW	8)HRF GF HW	9)HRF GFT HW	10)HRF GFTD HW	11)HRF media dei parametri modificata	12)Spettro +lsqnonlin
1	42	AIC_medio	987.21	995.03	1000.84	987.45	988.49	988.29	987.02	995.43	996.10
		n.voxel attivi	451	273	220	447	354	378	455	168	175
2	54	AIC_medio	1061.58	1064.86	1066.12	1062.04	1064.05	1061.30	1060.08	1059.02	1063.16
		n.voxel attivi	283	274	205	280	304	320	249	264	268
3	91	AIC_medio	973.57	971.93	973.28	973.69	970.66	968.50	969.57	974.31	972.79
		n.voxel attivi	222	305	251	220	272	273	231	209	230
4	85	AIC_medio	1037.82	1042.27	1046.81	1040.25	1039.52	1037.27	1037.66	1036.17	1036.74
		n.voxel attivi	459	695	199	438	544	559	348	483	433
5	9	AIC_medio	1049.70	1051.06	1049.44	1050.87	1050.55	1050.72	1051.15	1049.34	1051.05
		n.voxel attivi	61	75	41	68	79	77	52	75	45
6	333	AIC_medio	1028.32	1036.28	1035.87	1035.97	1041.12	1033.51	1032.39	1030.17	1030.01
		n.voxel attivi	1373	1066	1181	1074	951	1040	1045	1322	1335
7	0	AIC_medio	-----	-----	-----	-----	-----	-----	-----	-----	-----
		n.voxel attivi	12	6	7	14	9	9	11	7	30
8	81	AIC_medio	1081.06	1087.72	1085.44	1081.32	1086.48	1082.04	1080.8	1089.95	1089.33
		n.voxel attivi	691	499	397	686	569	605	624	397	340
9	0	AIC_medio	-----	-----	-----	-----	-----	-----	-----	-----	-----
		n.voxel attivi	43	35	27	42	34	38	31	25	118
10	725	AIC_medio	1039.51	1048.47	1066.81	1040.02	1039.06	1034.24	1031.01	1044.36	1043.84
		n.voxel attivi	3367	4450	1138	2944	3498	3860	3690	3332	3219
11	2	AIC_medio	1094.04	1106.54	1104.60	1094.41	1101.18	1095.65	1090.9	1104.44	1088.93
		n.voxel	261	115	318	256	212	231	208	93	118

Soggetti	n. voxel comuni		1-4)HRF BV migliore	5)HRF media dei parametri	6)HRF media pesata dei parametri	7)HRF B HW	8)HRF GF HW	9)HRF GFT HW	10)HRF GFTD HW	11)HRF media dei parametri modificata	12)Spettro +lsqnonlin
		attivi									
12	461	AIC_medio	1170.18	1169.59	1170.82	1170.91	1171.48	1171.35	1171.12	1168.61	1168.5
		n.voxel attivi	733	772	777	713	678	704	799	875	869
13	16	AIC_medio	1062.63	1056.18	1061.16	1062.58	1060.30	1061.07	1061.67	1056.33	1056.77
		n.voxel attivi	100	193	65	100	134	135	93	234	197
14	255	AIC_medio	1137.86	1144.75	1146.74	1142.66	1147.40	1142.07	1140.43	1144.47	1140.29
		n.voxel attivi	937	874	746	907	850	908	947	886	928
15	313	AIC_medio	1157.63	1164.31	1163.13	1157.83	1161.73	1158.67	1156.81	1160.05	1156.22
		n.voxel attivi	1182	1161	696	1176	1219	1259	1144	1155	1132
16	9	AIC_medio	1036.58	1047.04	1065.03	1036.81	1037.26	1037.47	1035.93	1044.31	1035.90
		n.voxel attivi	156	160	33	158	183	184	132	153	185
17	174	AIC_medio	1088.82	1091.23	1096.87	1089.04	1090.25	1090.57	1089.83	1089.48	1098.31
		n.voxel attivi	1177	942	1063	1169	1028	1089	1133	1013	660
18	1670	AIC_medio	743.05	743.82	752.40	742.09	742.62	740.92	742.17	745.76	744.03
		n.voxel attivi	2893	2868	2194	2920	2956	2998	1891	2635	2741
19	31	AIC_medio	944.13	948.35	954.87	946.33	942.85	943.36	942.95	945.08	967.83
		n.voxel attivi	200	247	118	170	221	227	66	181	53
20	112	AIC_medio	1042.38	1052.07	1054.60	1042.65	1045.26	1044.30	1043.11	1047.04	1049.38
		n.voxel attivi	524	339	291	519	476	486	473	405	372

```

% CODICE DI CONFRONTO DEGLI AIC_medi SUI VOXEL COMUNI
% Confronto degli AIC delle 4 HRF BV_QXtool per scegliere l'hrf BV migliore
1) nR=64;
2) nC=64;
3) nfette=36;
4) v_a1=zeros(nR,nC,nfette); % 1)selezionati boynton
5) v_a2=zeros(nR,nC,nfette); % 2)selezionati two-gamma
6) v_a3=zeros(nR,nC,nfette); % 3)selezionati two-gamma-der
7) v_a4=zeros(nR,nC,nfette); % 4)selezionati two-gamma-der-disp
    %1)
8) for i=1:length(selezione1),
9)     r=selezione1(i,1);
10)    c=selezione1(i,2);
11)    f=selezione1(i,3);
12)    v_a1(r,c,f)=1;
13) end
    %2)
14) for i=1:length(selezione2),
15)     r=selezione2(i,1);
16)     c=selezione2(i,2);
17)     f=selezione2(i,3);
18)     v_a2(r,c,f)=1;
19) end
    %3)
20) for i=1:length(selezione3),
21)     r=selezione3(i,1);
22)     c=selezione3(i,2);
23)     f=selezione3(i,3);
24)     v_a3(r,c,f)=1;
25) end
    %4)
26) for i=1:length(selezione4),
27)     r=selezione4(i,1);
28)     c=selezione4(i,2);
29)     f=selezione4(i,3);
30)     v_a4(r,c,f)=1;
31) end
32) v_a_t=v_a1+v_a2+v_a3+v_a4; % i voxel attivi comuni valgono 4
33) v_a_c=(v_a_t==4); % matrice con 1 sui voxel comuni
34) n_v_c=sum(sum(sum(v_a_c))); % numero dei voxel attivi comuni
    % AIC1=AIC della Boynton
    % AIC2=AIC della two-gamma
    % AIC3=AIC della two-gamma-der
    % AIC4=AIC della two-gamma-der-disp
35) AIC1_medio=sum(sum(sum(AIC1.*v_a_c)))/n_v_c; % AIC medio sui voxel
comuni      della boynton
36) AIC2_medio=sum(sum(sum(AIC2.*v_a_c)))/n_v_c; % AIC medio sui voxel
comuni      della two-gamma
37) AIC3_medio=sum(sum(sum(AIC3.*v_a_c)))/n_v_c; % AIC medio sui voxel
comuni      della two-gamma-der

```

Cod.6.1a: Codice relativo al primo confronto


```

38) AIC4_medio=sum(sum(sum(AIC4.*v_a_c))/n_v_c; % AIC medio sui voxel
comuni della two-      gamma-der-disp
39) AIC_med_4=[AIC1_medio AIC2_medio AIC3_medio AIC4_medio];
40) AIC_migl=min(AIC_med_4);
41) disp(['AIC1_medio= ',num2str(AIC1_medio),' AIC2_medio=
',num2str(AIC2_medio),'      AIC3_medio= ',num2str(AIC3_medio),' AIC4_medio=
',num2str(AIC4_medio),' AIC  migliore= ',num2str(AIC_migl)])
42) disp(['selezione1= ',num2str(length(selezione1)),' selezione2=
',num2str(length(selezione2)),'      selezione3= ',num2str(length(selezione3)),'
selezione4= ',num2str(length(selezione4))])
      % Confronto tra il migliore degli AIC delle 4 HRF BV_QXtool con gli AIC degli
altri codici per scegliere l'hrf BV migliore
43) if AIC_migl==AIC1_medio,
44)  selezione=selezione1;
45)  AIC=AIC1;
46) end
47) if AIC_migl==AIC2_medio,
48)  selezione=selezione2;
49)  AIC=AIC2;
50) end
51) if AIC_migl==AIC3_medio,
52)  selezione=selezione3;
53)  AIC=AIC3;
54) end
55) if AIC_migl==AIC4_medio,
56)  selezione=selezione4;
57)  AIC=AIC4;
58) end
59) v_a=zeros(nR,nC,nfette); % 1-4) voxel di selezione
60) v_a5=zeros(nR,nC,nfette); % 5) voxel di selezione5 (hrf media dei parametri)
61) v_a6=zeros(nR,nC,nfette); % 6) voxel di selezione6 (hrf media pesata dei
parametri)
62) v_a7=zeros(nR,nC,nfette); % 7) voxel di selezione7 (hrf B Handwerker)
63) v_a8=zeros(nR,nC,nfette); % 8) voxel di selezione8 (hrf GF Handwerker)
64) v_a9=zeros(nR,nC,nfette); % 9) voxel di selezione9 (hrf GFTD HW)
65) v_a10=zeros(nR,nC,nfette); % 10) voxel di selezione10 (hrf GFTD HW)
66) v_a11=zeros(nR,nC,nfette); % 11) voxel di selezione11 (hrf media dei parametri
modificata)
67) v_a12=zeros(nR,nC,nfette); % 12) voxel di selezione12 (spettro+lsqnonlin)
      %1-4)
68) for i=1:length(selezione),
69)  r=selezione(i,1);
70)  c=selezione(i,2);
71)  f=selezione(i,3);
72)  v_a(r,c,f)=1;
73) end
      %5)
74) for i=1:length(selezione5),
75)  r=selezione5(i,1);
76)  c=selezione5(i,2);
77)  f=selezione5(i,3);
78)  v_a5(r,c,f)=1;
79) end

```

Cod. 6. 1b: Codice relativo al primo confronto.

```
%6)
80) for i=1:length(selezione6),
81)   r=selezione6(i,1);
82)   c=selezione6(i,2);
83)   f=selezione6(i,3);
84)   v_a6(r,c,f)=1;
85) end
%7)
86) for i=1:length(selezione7),
87)   r=selezione7(i,1);
88)   c=selezione7(i,2);
89)   f=selezione7(i,3);
90)   v_a7(r,c,f)=1;
91) end
%8)
92) for i=1:length(selezione8),
93)   r=selezione8(i,1);
94)   c=selezione8(i,2);
95)   f=selezione8(i,3);
96)   v_a8(r,c,f)=1;
97) end
%9)
98) for i=1:length(selezione9),
99)   r=selezione9(i,1);
100)  c=selezione9(i,2);
101)  f=selezione9(i,3);
102)  v_a9(r,c,f)=1;
103) end
%10)
104) for i=1:length(selezione10),
105)   r=selezione10(i,1);
106)   c=selezione10(i,2);
107)   f=selezione10(i,3);
108)   v_a10(r,c,f)=1;
109) end
%11)
110) for i=1:length(selezione11),
111)   r=selezione11(i,1);
112)   c=selezione11(i,2);
113)   f=selezione11(i,3);
114)   v_a11(r,c,f)=1;
115) end
%12)
116) for i=1:length(selezione12),
117)   r=selezione12(i,1);
118)   c=selezione12(i,2);
119)   f=selezione12(i,3);
120)   v_a12(r,c,f)=1;
121) end
```

Cod.6.1c: Codice relativo al primo confronto.

```

122) v_a_tot=v_a+v_a5+v_a6+v_a7+v_a8+v_a9+v_a10+v_a11+v_a12; % i
voxel attivi      comuni valgono 4
123) v_a_com=(v_a_tot==9); % matrice con 1 sui voxel comuni
124) n_v_com=sum(sum(sum(v_a_com))); % numero dei voxel attivi comuni
      % AIC medi per ogni codice
125) AIC_migl_m=sum(sum(sum(AIC.*v_a_com)))/n_v_com; % AIC medio sui
voxel      comuni della migliore del prog. vecchio
126) AIC5_medio=sum(sum(sum(AIC5.*v_a_com)))/n_v_com;
127) AIC6_medio=sum(sum(sum(AIC6.*v_a_com)))/n_v_com;
128) AIC7_medio=sum(sum(sum(AIC7.*v_a_com)))/n_v_com;
129) AIC8_medio=sum(sum(sum(AIC8.*v_a_com)))/n_v_com;
130) AIC9_medio=sum(sum(sum(AIC9.*v_a_com)))/n_v_com;
131) AIC10_medio=sum(sum(sum(AIC10.*v_a_com)))/n_v_com;
132) AIC11_medio=sum(sum(sum(AIC11.*v_a_com)))/n_v_com;
133) AIC12_medio=sum(sum(sum(AIC12.*v_a_com)))/n_v_com;

134) disp(['AIC_migl_m4= ',num2str(AIC_migl_m),' AIC5_medio=
        ',num2str(AIC5_medio),' AIC6_medio= ',num2str(AIC6_medio),'
AIC7_medio= ',num2str(AIC7_medio),' AIC8_medio= ',num2str(AIC8_medio),'
AIC9_medio= ',num2str(AIC9_medio),' AIC10_medio= ',num2str(AIC10_medio),'
AIC11_medio= ',num2str(AIC11_medio),' AIC12_medio=
        ',num2str(AIC12_medio)])
135) disp(['selezione= ',num2str(length(selezione)), 'selezione5=
        ',num2str(length(selezione5)), 'selezione6= ',num2str(length(selezione6)),
selezione7= ',num2str(length(selezione7)), 'selezione8=
        ',num2str(length(selezione8)), 'selezione9= ',num2str(length(selezione9)),
selezione10= ',num2str(length(selezione10)), 'selezione11=
        ',num2str(length(selezione11)), 'selezione12=
        ',num2str(length(selezione12))])

```

Cod.6.1d: Codice relativo al primo confronto.

6.2 Gruppo A

Intendiamo per *Gruppo A* tutti i 4 metodi ad hrf fissa considerati nel capitolo 4. Si vuole capire quale di questi metodi applicato al data set sia il migliore; quindi si è calcolato l'indice AIC medio sui voxel comuni ai quattro metodi. Infine si è scelto come metodo migliore quello con indice di AIC minore. Il codice Matlab® che permette tale confronto per questi quattro metodi è riportato in Cod.6.2. Nelle righe 1-4 si sono inizializzate le matrici che utilizzeremo in seguito; nR e nC sono il numero di righe e colonne rispettivamente di una fetta del soggetto e sono entrambe pari a 64, nfette sono il numero di fette di ciascun soggetto e sono 36. Dalla riga 5 alla 28 si creano 4 maschere una per ogni metodo in cui sono posti a 1 i voxel selezionati dal metodo corrispondente e a 0 tutti gli altri. In *selezione7* sono riportate le posizioni (*r*, *c*, *f* sono riga, colonna e fetta) dei voxel risultati attivi applicando il metodo GLM illustrato nel capitolo 4 in cui si è utilizzata come *h(t)* la single gamma

HW; *AIC7* è la matrice degli indici di AIC relativi ai voxel selezionati. Analogamente in *selezione8* sono riportati i voxel selezionati sempre con il GLM e

```
%AIC_medio sui voxel comuni al Gruppo A
1-4).....CODICE OMESSO righe 62-65 Fig.6.1b.....
%7)
5-28).....CODICE OMESSO righe 86-109 Fig.6.1c .....
1) v_a_tot2=v_a7+v_a8+v_a9+v_a10; % i voxel attivi comuni valgono 4
2) v_a_com2=(v_a_tot2==4); % matrice con 1 sui voxel comuni
3) n_v_com2=sum(sum(sum(v_a_com2))); % numero dei voxel attivi comuni
% AIC medi per ogni codice
4) AIC7_medio=sum(sum(sum(AIC7.*v_a_com2)))/n_v_com2;
5) AIC8_medio=sum(sum(sum(AIC8.*v_a_com2)))/n_v_com2;
6) AIC9_medio=sum(sum(sum(AIC9.*v_a_com2)))/n_v_com2;
7) AIC10_medio=sum(sum(sum(AIC10.*v_a_com2)))/n_v_com2;

8) disp([' AIC7_medio= ',num2str(AIC7_medio),' AIC8_medio=
',num2str(AIC8_medio),' AIC9_medio= ',num2str(AIC9_medio),' AIC10_medio=
',num2str(AIC10_medio)])
9) disp([' selezione7= ',num2str(length(selezione7)),' selezione8=
',num2str(length(selezione8)),' selezione9= ',num2str(length(selezione9)),'
selezione10= ',num2str(length(selezione10))])
```

Cod.6.2: Codice che calcola l'indice di Akaike medio per i quattro metodi del gruppo A.

utilizzando come $h(t)$ la hrf two-gamma HW. *AIC8* è la matrice degli indici di AIC relativi a tale selezione. In *selezione9* e *10* sono riportati, come sopra, i voxel attivi ottenuti rispettivamente con i metodi GFT e GFTD visti nel paragrafo 4.2 e *AIC9* e *AIC10* sono le relative matrici di AIC. I risultati ottenuti per tutti i 20 soggetti sono riportati in tabella 6.2, dove il risultato con AIC medio minore è evidenziato in grassetto. Dalla tabella si evince che i metodi 7 e 10 danno il risultato migliore su 6 soggetti ciascuno, mentre per quanto riguarda i rimanenti 2: il metodo 8 risulta il migliore su 3 soggetti, e il metodo 9 su 5.

Tab.6.2: Risultati ottenuti relativi al Gruppo A.

Soggetti	n.voxel comuni		7)HRF B HW	8)HRF GF HW	9)HRF GFT HW	10)HRF GFTD HW
1	274	AIC_medio	999.18	999.76	999.16	998.79
		n.voxel attivi	447	354	378	455
2	169	AIC_medio	1056.92	1056.77	1055.71	1055.82
		n.voxel attivi	280	304	320	249
3	155	AIC_medio	975.40	973.98	972.79	973.37
		n.voxel attivi	220	272	273	231
4	277	AIC_medio	1033.01	1031.10	1028.79	1028.99
		n.voxel attivi	438	544	559	348

Soggetti	n.voxel comuni		7)HRF B HW	8)HRF GF HW	9)HRF GFT HW	10)HRF GFTD HW
5	28	AIC_medio	1067.33	1066.73	1066.86	1067.29
		n.voxel attivi	68	79	77	52
6	717	AIC_medio	1034.93	1036.69	1032.86	1032.84
		n.voxel attivi	1074	951	1040	1045
7	4	AIC_medio	1041.18	1040.85	1040.89	1040.72
		n.voxel attivi	14	9	9	11
8	448	AIC_medio	1071.49	1073.07	1071.89	1072.03
		n.voxel attivi	686	569	605	624
9	24	AIC_medio	1072.23	1073.29	1073.26	1074.33
		n.voxel attivi	42	34	38	31
10	2700	AIC_medio	1050.58	1045.90	1032.42	1030.69
		n.voxel attivi	2944	3498	3860	3690
11	142	AIC_medio	1105.41	1105.88	1105.45	1105.72
		n.voxel attivi	256	212	231	208
12	557	AIC_medio	1171.70	1171.98	1171.98	1171.98
		n.voxel attivi	713	678	704	799
13	54	AIC_medio	1026.76	1025.46	1025.54	1025.92
		n.voxel attivi	100	134	135	93
14	688	AIC_medio	1116.85	1117.85	1115.85	1115.68
		n.voxel attivi	907	850	908	947
15	884	AIC_medio	1153.87	1154.80	1153.81	1153.50
		n.voxel attivi	1176	1219	1259	1144
16	96	AIC_medio	1082.71	1081.88	1080.55	1080.64
		n.voxel attivi	158	183	184	132
17	832	AIC_medio	1078.56	1079.60	1078.69	1078.66
		n.voxel attivi	1169	1028	1089	1133
18	1849	AIC_medio	740.62	740.96	739.28	740.52
		n.voxel attivi	2920	2956	2998	1891
19	58	AIC_medio	928.89	926.82	926.86	927.10
		n.voxel attivi	170	221	227	66
20	369	AIC_medio	1046.87	1047.81	1047.46	1047.40
		n.voxel attivi	519	476	486	473

6.3 Gruppo B

Nel *Gruppo B* sono raggruppati i 4 metodi sviluppati nel capitolo 5, ossia hrf con media dei parametri, hrf con media pesata dei parametri, hrf con media dei parametri modificata e hrf costruita con lo spettro di potenza. Per agevolarne la distinzione sono stati contrassegnati rispettivamente con i numeri 5, 6, 11 e 12. La scelta del metodo migliore è stata sviluppata come per il *Gruppo A* utilizzando come discriminante l'indice AIC medio minore. Il codice relativo è riportato in Cod.6.3 e come struttura è del tutto simile a quello relativo al *Gruppo A*; in questo caso le

```

%AIC_medio sui voxel comuni al Gruppo3
.....CODICE OMESSO righe 60,61,66,67 Fig.6.1b.....
%5)
.....CODICE OMESSO righe 74-85 Fig.6.1b-6.1c.....
%11)
.....CODICE OMESSO righe 110-121 Fig.6.1c.....
29) v_a_tot3=v_a5+v_a6+v_a11+v_a12;
30) v_a_com3=(v_a_tot3==4); % matrice con 1 sui voxel comuni
31) n_v_com3=sum(sum(sum(v_a_com3))); % numero dei voxel attivi comuni
    % AIC medi per ogni codice
32) AIC5_medio=sum(sum(sum(AIC5.*v_a_com3)))/n_v_com3;
33) AIC6_medio=sum(sum(sum(AIC6.*v_a_com3)))/n_v_com3;
34) AIC11_medio=sum(sum(sum(AIC11.*v_a_com3)))/n_v_com3;
35) AIC12_medio=sum(sum(sum(AIC12.*v_a_com3)))/n_v_com3;

36) disp([' AIC5_medio= ',num2str(AIC5_medio),' AIC6_medio=
        ',num2str(AIC6_medio),' AIC11_medio= ',num2str(AIC11_medio),'
        AIC12_medio= ',num2str(AIC12_medio)])
37) disp([' selezione5= ',num2str(length(selezione5)),' selezione6=
        ',num2str(length(selezione6)),' selezione11= ',num2str(length(selezione11)),'
        selezione12= ',num2str(length(selezione12))])

```

Cod.6.3: Codice che calcola l'indice di Akaike medio per i quattro metodi del gruppo B.

matrici con le posizioni dei voxel sono *selezione5*, *selezione6*, *selezione11* e *selezione12*. Allo stesso modo anche le matrici degli indici AIC sono state chiamate *AIC5*, *AIC6*, *AIC11* e *AIC12*. In tabella 6.3 sono riportati il numero di voxel comuni ai quattro metodi, l'indice AIC medio calcolato solo sui voxel comuni e il numero di voxel selezionati da ciascun metodo.

Si noti che per i soggetti 7 e 9 non essendovi voxel comuni ai quattro metodi non è stato possibile il calcolo dell'indice AIC medio, pertanto non sono stati presi in considerazione e di conseguenza il data set si riduce a 18 soggetti. Il metodo 5 risulta il migliore per 3 soggetti, il 6 per un solo soggetto, i metodi 11 e 12 risultano i migliori in 7 casi ciascuno.

Tab.6.3: Risultati ottenuti relativi al Gruppo B.

Soggetti	n.voxel comuni		5)HRF media dei parametri	6)HRF media pesata dei parametri	11)HRF media dei parametri modificata	12)Spettro +lsqnonlin
1	42	AIC_medio	995.03	1000.84	995.43	996.10
		n.voxel attivi	273	220	168	175
2	56	AIC_medio	1062.53	1063.75	1056.73	1060.81
		n.voxel attivi	274	205	264	268
3	131	AIC_medio	973.97	975.07	976.33	974.99
		n.voxel attivi	305	251	209	230
4	92	AIC_medio	1045.05	1049.46	1039.13	1039.57
		n.voxel attivi	695	199	483	433

Soggetti	n.voxel comuni		5)HRF media dei parametri	6)HRF media pesata dei parametri	11)HRF media dei parametri modificata	12)Spettro +lsqnonlin
5	11	AIC_medio	1055.08	1053.71	1053.37	1055.44
		n.voxel attivi	75	41	75	45
6	483	AIC_medio	1043.24	1039.03	1036.49	1034.29
		n.voxel attivi	1066	1181	1322	1335
7	0	AIC_medio	-----	-----	-----	-----
		n.voxel attivi	6	7	7	30
8	81	AIC_medio	1087.72	1085.44	1089.95	1089.33
		n.voxel attivi	499	397	397	340
9	0	AIC_medio	-----	-----	-----	-----
		n.voxel attivi	35	27	25	118
10	725	AIC_medio	1048.47	1066.81	1044.36	1043.84
		n.voxel attivi	4450	1138	3332	3219
11	2	AIC_medio	1106.54	1104.60	1104.44	1088.93
		n.voxel attivi	151	318	93	118
12	591	AIC_medio	1164.61	1165.02	1163.45	1163.36
		n.voxel attivi	772	777	875	869
13	31	AIC_medio	1038.34	1042.76	1038.28	1039.18
		n.voxel attivi	193	65	234	197
14	313	AIC_medio	1144.03	1143.39	1143.64	1139.14
		n.voxel attivi	874	746	886	928
15	319	AIC_medio	1164.48	1163.30	1160.22	1156.45
		n.voxel attivi	1161	696	1155	1132
16	9	AIC_medio	1047.04	1065.03	1044.31	1035.90
		n.voxel attivi	160	33	153	185
17	174	AIC_medio	1091.23	1096.87	1089.48	1098.31
		n.voxel attivi	942	1063	1013	660
18	2025	AIC_medio	744.41	752.16	746.27	744.78
		n.voxel attivi	2868	2194	2635	2741
19	34	AIC_medio	951.07	956.76	947.81	969.03
		n.voxel attivi	247	118	181	53
20	112	AIC_medio	1052.07	1054.60	1047.04	1049.38
		n.voxel attivi	339	291	405	372

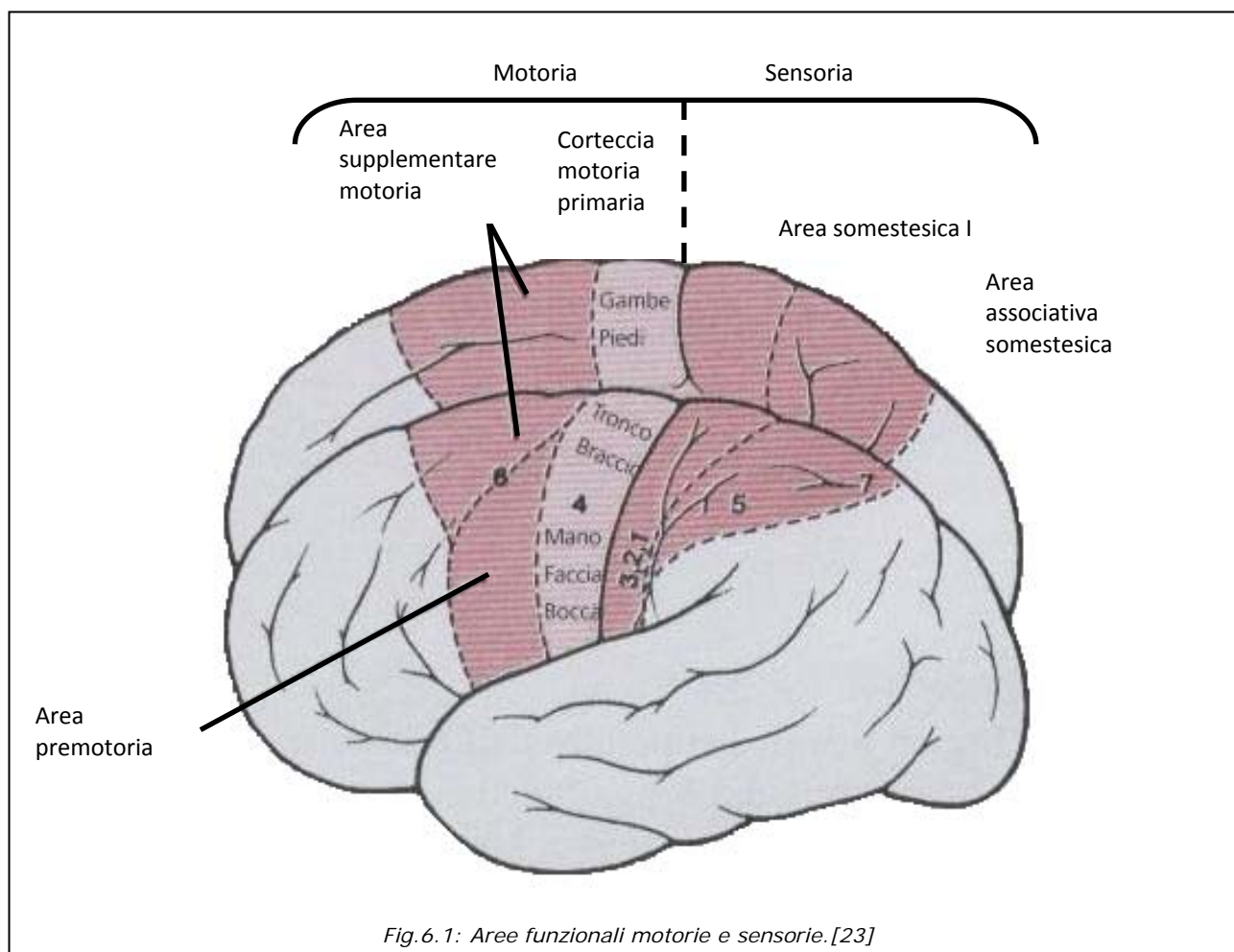
6.4 Confronto dei risultati ottenuti con i metodi ottimi

Dall'analisi dei risultati di tabella 6.1 risulta che i metodi migliori del Gruppo A sono il 7 e il 10 mentre del Gruppo B sono l'11 e 12. Sempre da questa tabella, se si fa un confronto sugli indici AIC medi, solo per questi 4 gruppi si nota che il metodo 7 risulta migliore su 2 soggetti sani e 1 paziente. Il metodo 10 su 1 sano e 3 pazienti; il metodo 11 su 4 pazienti e il metodo 12 su 6 pazienti. Dai risultati possiamo dedurre

che i metodi del Gruppo A si adattano meglio a soggetti sani, come era logico attendersi in quanto l'hrf scelta è standard. I metodi del Gruppo B risultano migliori per i soggetti colpiti da stroke in quanto l'hrf media specifica per il soggetto è stata adattata al segnale BOLD, che in questa categoria di pazienti ha una risposta emodinamica ritardata rispetto a quella di un soggetto sano. Per la validità di queste considerazioni va comunque tenuto conto che nel nostro data set i soggetti sani sono solamente 3 quindi un gruppo troppo esiguo rispetto ai 14 pazienti di cui è stato possibile fare un'analisi.

6.5 Analisi dei risultati in rapporto alla gravità della lesione nei diversi soggetti

Dopo aver analizzato i risultati dal punto di vista numerico, si è deciso di procedere anche ad un confronto visivo tra i metodi, riscontrando se effettivamente tali metodi rilevano delle zone di attivazione che correlano con la clinica dei pazienti affetti da stroke. In figura 6.1 sono riportate le aree che si attivano durante il movimento della

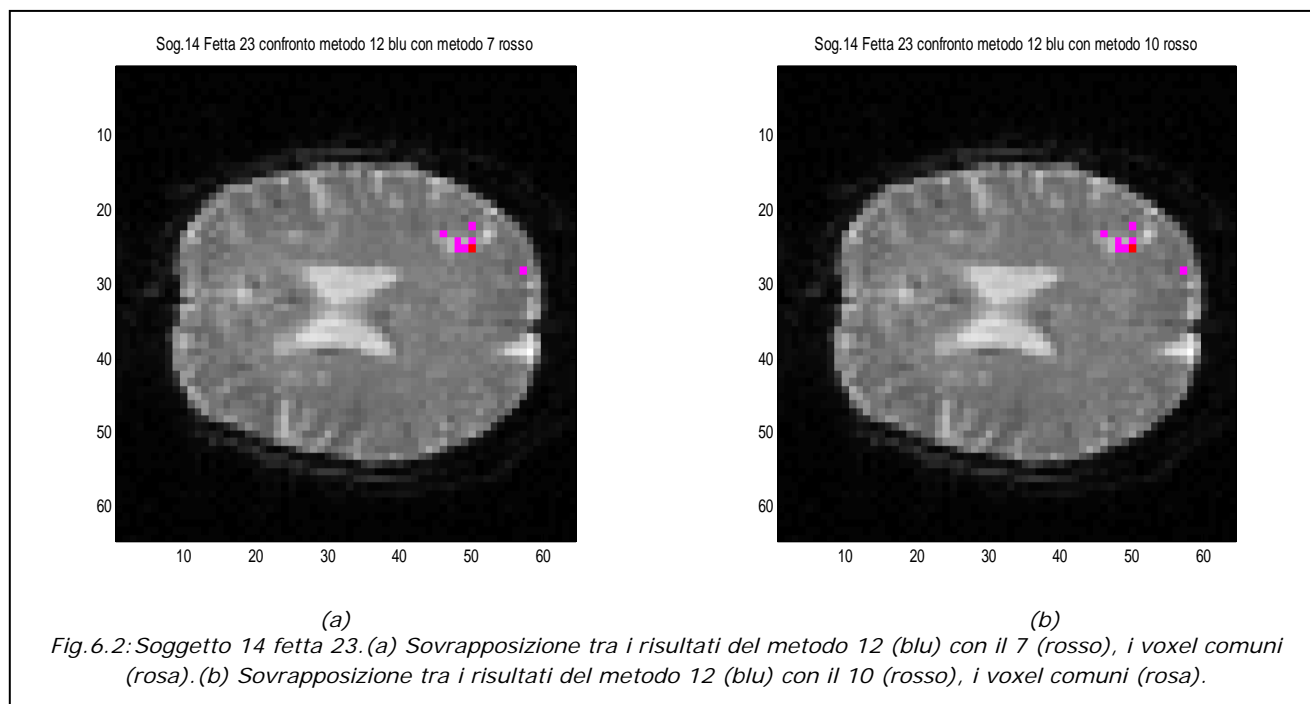


mano in un generico soggetto. Va considerato che, essendo la morfologia del cervello diversa da soggetto a soggetto, tale immagine è solo indicativa. È di interesse particolare per questa tesi riuscire infine a decidere quale metodo dei quattro è auspicabile utilizzare su un qualsiasi nuovo paziente.

6.5.1 Analisi visiva dei risultati

Dall'analisi visiva dei risultati si è riscontrato che in qualche caso il metodo che risulta migliore secondo l'indice AIC non sembra esserlo visivamente. Quindi si è ritenuto opportuno controllare i risultati anche per i metodi non ottimi del Gruppo A e del Gruppo B. Si sono analizzati i 20 soggetti del data set, escluso il 10 per i motivi già spiegati, si è riscontrato che tutti i soggetti presentano delle attivazioni ad eccezione dei soggetti 5, 7 e 9. Per quanto riguarda il 7 e il 9 non si è riusciti a calcolare l'indice di AIC medio in quanto non presentano voxel comuni; inoltre questi soggetti presentavano il primo una lesione di tipo LACI e il secondo una lesione di tipo TACI (tabella 2.2). Nel primo caso dalla scala NIH e dall'indice di Barthel si deduce che tale soggetto non ha recuperato a distanza di tempo; nel secondo caso il PEM è assente ed inoltre la scala NIH e l'indice di Barthel relativi indicano una situazione ancora più difficile con recupero pressoché inesistente. Da tali considerazioni è logico attendersi che questi due pazienti non presentino attivazione. Il soggetto 5 contrariamente alle informazioni forniteci dalla tabella 2.2 presenta voxel sparsi con tutti i metodi. Per tutti gli altri soggetti vi sono invece delle zone di attivazione che verranno in seguito discusse.

Nel soggetto 12 i voxel significativamente attivi formano una corona posizionata nella parte più esterna della maschera, tale attivazione non è fisiologicamente interpretabile e molto probabilmente è legata al movimento della testa pertanto il soggetto non presenta reali attivazioni. Per il soggetto 13 non è riportato in tabella 2.2 il lato della lesione; dalla mappa statistica dei risultati si nota che attiva poco a sinistra con i metodi 11 e 12 mentre non attiva affatto con gli altri. Il soggetto 14 contrariamente alle aspettative di tabella 2.2 riporta un'attivazione dell'insula destra, ossia ipsi-laterale allo stimolo, con tutti i metodi. Tale attivazione relativa alle fette 23 e 24 è visualizzata in figura 6.2 e 6.3 in cui si sono confrontati il metodo 12, ossia quello con AIC medio minore, con i metodi migliori dei gruppi A e B. Si nota la



mancanza di voxel di colore blu in quanto i voxel rilevati dal metodo 12 sono attivi anche per gli altri metodi.

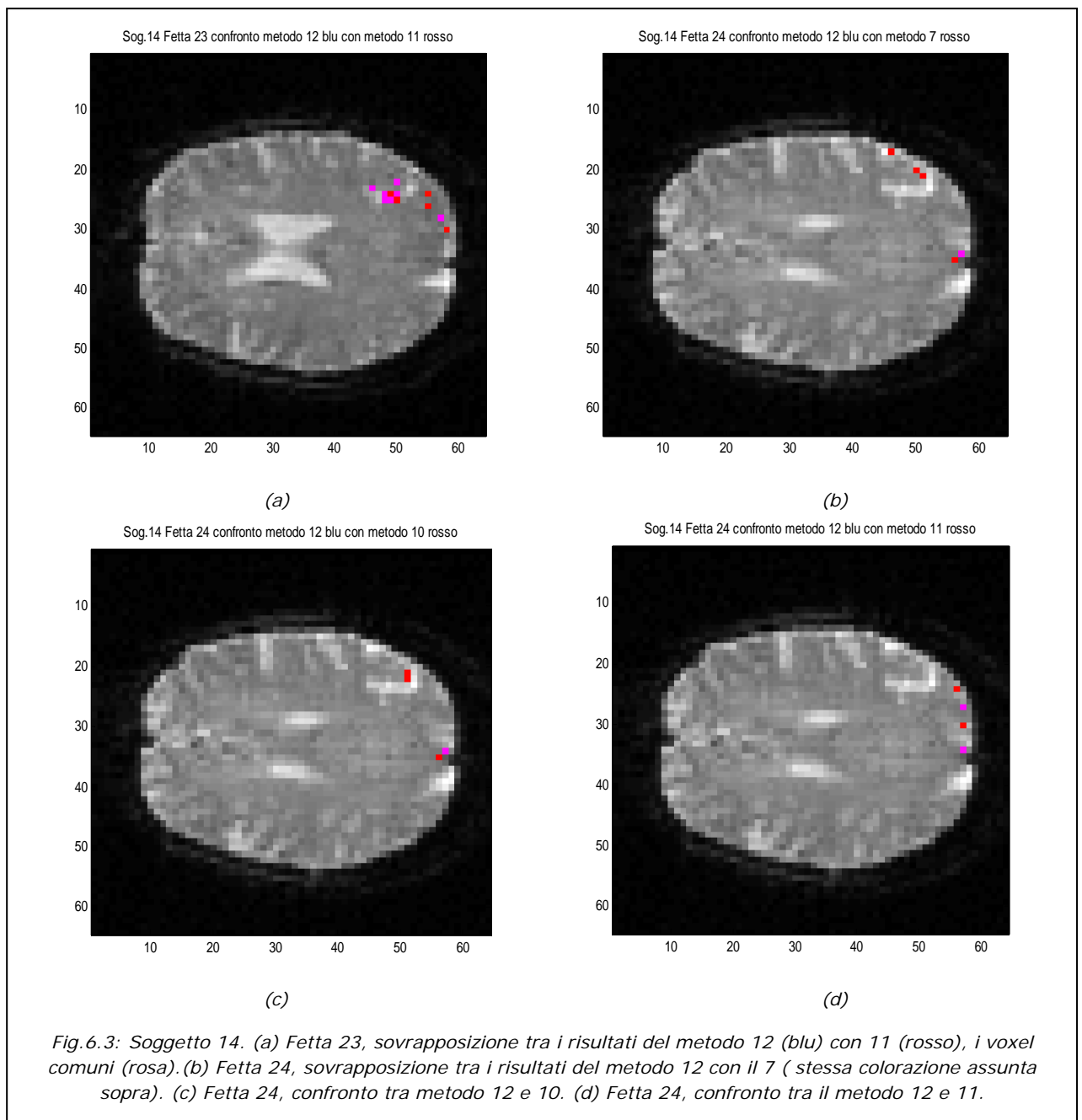
Nel soggetto 17 vi è la presenza di un'attivazione eccessiva nell'area SI in rapporto ai dati riportati in tabella 2.2 forse causata dall'eccessivo rumore che affligge i dati; ad ogni modo in questo caso il metodo che dà minore attivazione è il 12. In tale soggetto l'attivazione si presenta nelle fette 29, 30 e 31 ed è confermata da tutti i metodi.

I pazienti che danno attivazione migliore sono l'1, 2, 3, 4, 6, 8, 11, 16 e i tre soggetti sani.

Il paziente 1, come ci si poteva aspettare dalla tabella 2.2, presenta un'attivazione dell'area SI con tutti i quattro metodi le fette in cui è presente l'attivazione sono la 20, 21 e 22 e la 26, 27, 28, 29, 30. A titolo di esempio si è voluto riportare in figura 6.4 (a) e 6.5 (a) il confronto tra il metodo migliore in assoluto (metodo 10) secondo l'AIC (tabella 6.1), con quello peggiore (metodo 6), che non faceva parte dei metodi ottimi. Nel seguito, visti gli scarsi risultati del metodo 6 anche per gli altri soggetti, si è scelto di non riportarlo nei confronti. Inoltre dai confronti delle fig. 6.4 e 6.5 si nota una sostanziale equivalenza degli altri metodi al più con qualche voxel di differenza; in particolare nella zona di interesse della fetta 28 il metodo 7 salta 2 voxel attivi rispetto al 10; il metodo 11 non rileva 1 voxel attivo; infine il metodo 12, come il precedente, non ne rileva 1, in compenso aggiunge un voxel attivo. Per quanto

riguarda la zona d'interesse della fetta 29 il metodo 7 salta 4 voxel, i metodi 11 e 12 non ne rilevano 2. Quindi per tale soggetto il metodo peggiore dei 4 è il 7, mentre il migliore può essere scelto tra il 10, 11 e 12.

Nonostante il soggetto 2 presenti una lesione TACI a sinistra e il PEM ridotto, si ha un recupero temporale fino all'autosufficienza e l'attivazione conferma tale recupero: si nota infatti l'attivazione delle aree SI e SII controlaterali allo stimolo nelle fette 17, 18, 19, 20, 21, 22, 23, 24 con tutti i 4 metodi (figura 6.6). In questo caso il metodo 10 è quello che attiva di meno mentre il 12 è quello che presenta maggiore attivazione.



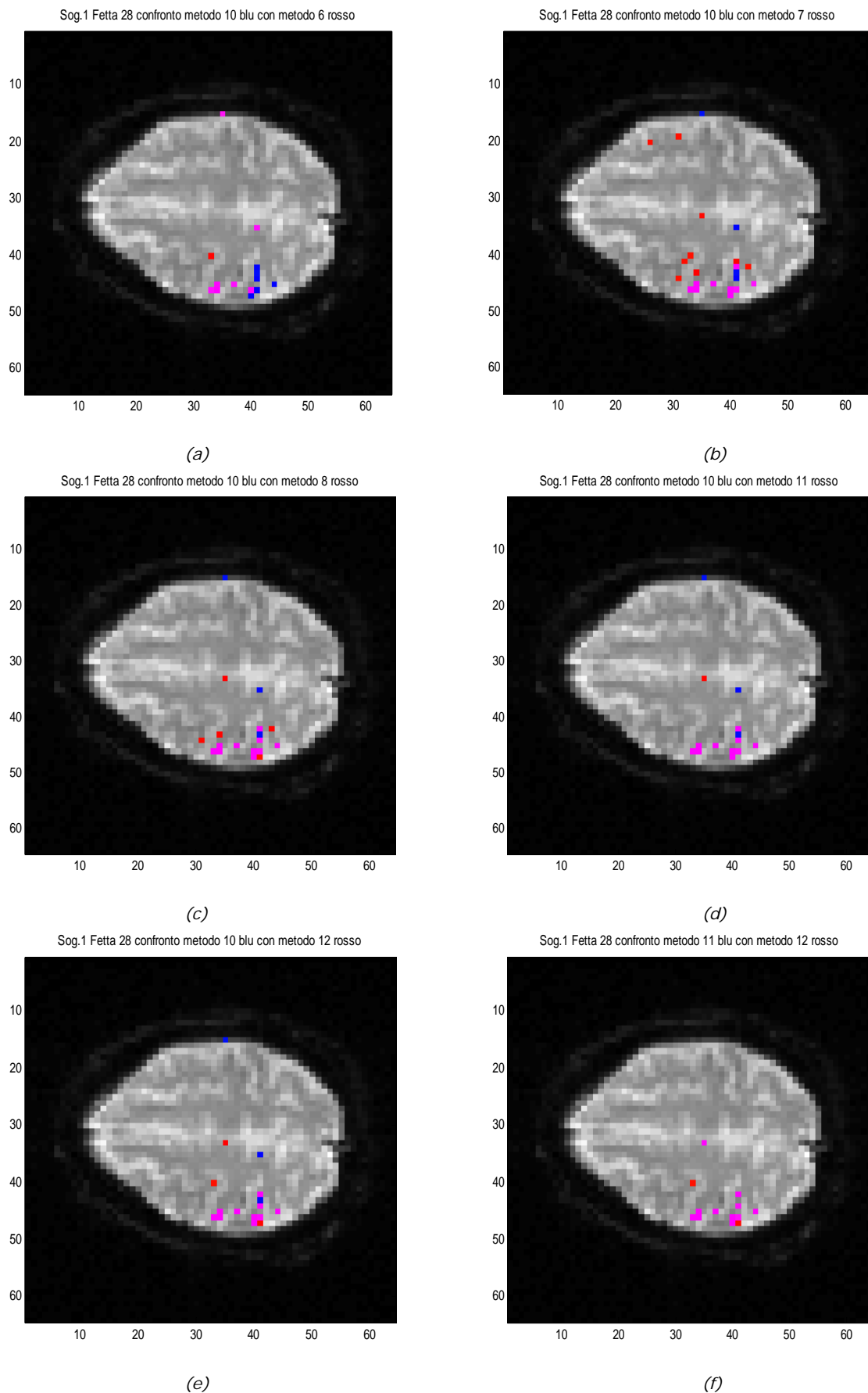
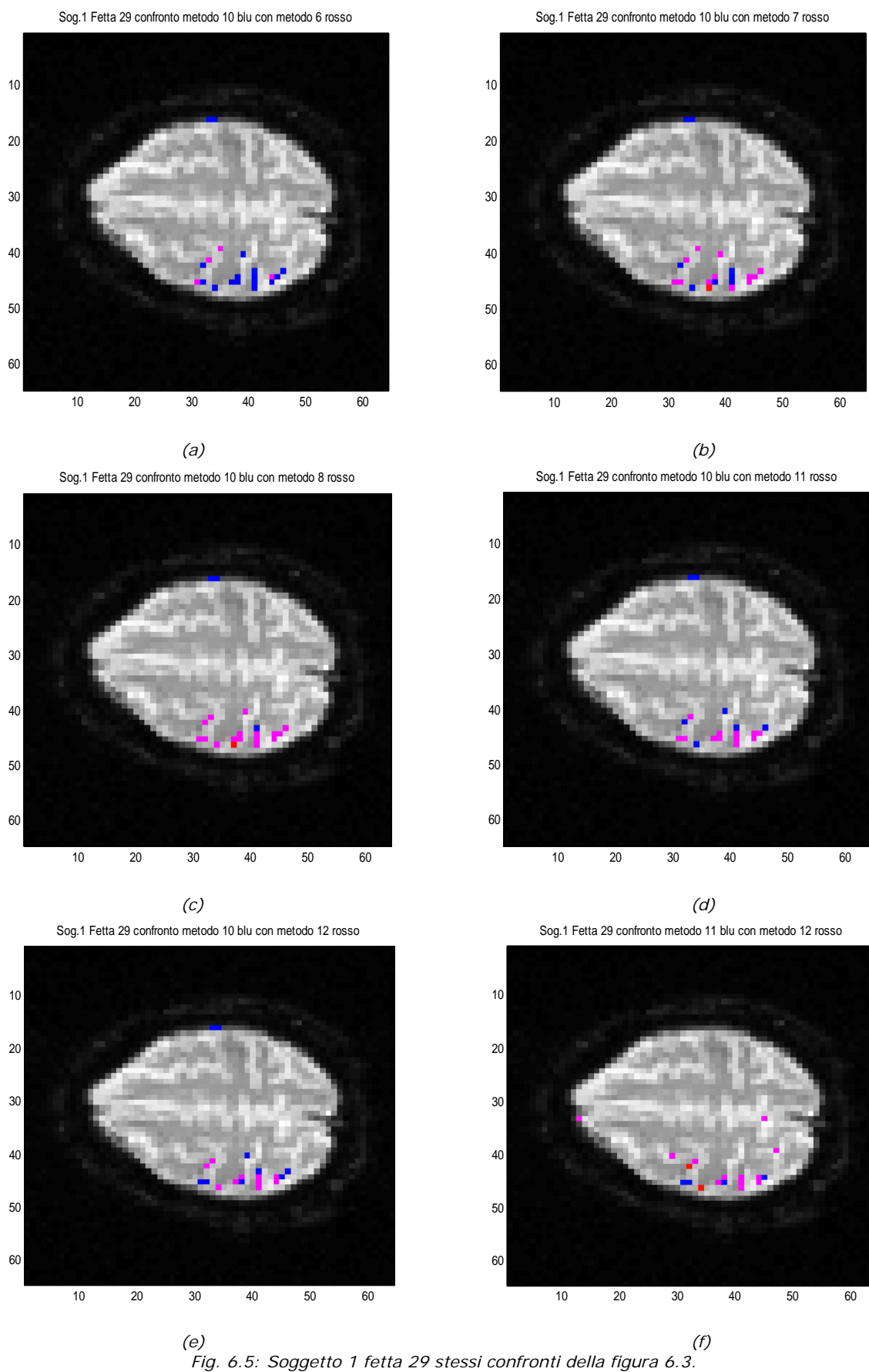


Fig.6.4: Soggetto 1 fetta 28 confronto tra: (a) metodo 10 e 6, (b) 10 e 7, (c) 10 e 8, (d) 10 e 11, (e) 10 e 12, (f) 11 e 12. Stessi colori della figura 6.2.



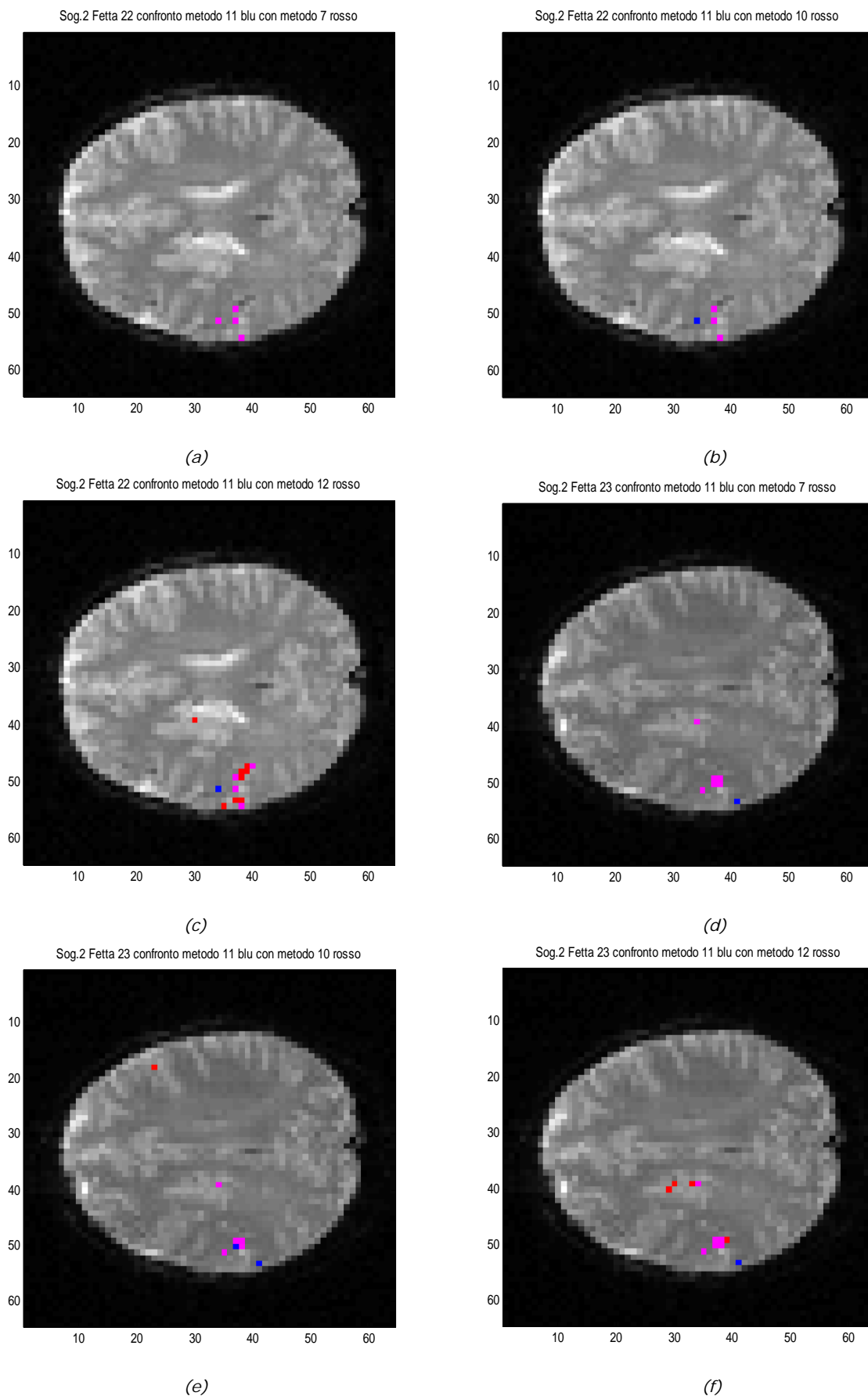


Fig.6.6: Soggetto 2: fetta 22 confronto tra (a) metodo 11 e 7, (b) 11 e 10, (c) 11 e 12. Gli stessi confronti (d), (e) ed (f) rispettivamente per la fetta 23.

Il soggetto 3 presenta un'attivazione dell'area SI di destra nelle fette dalla 27 alla 32 con tutti i 4 metodi (figure 6.7 e 6.8), il metodo visivamente migliore risulta il 10. I soggetti 4 e 6 attivano in modo piuttosto diffuso nella zona della lesione con tutti i metodi; tuttavia, se si confronta questa attivazione con i dati riportati in tabella, si evince un risultato poco verosimile che può essere spiegato dalla presenza di rumore che distorce i risultati. Il soggetto 8, che ha una lesione di tipo PACI a sinistra e un PEM ridotto, presenta scarsa attivazione nelle regioni di interesse. Il soggetto 11, con lesione a sinistra LACI e PEM ridotto, presenta un recupero quasi completo nel tempo. Si rileva un'attivazione a livello dell'insula controlaterale allo stimolo nelle fette 20, 21, 22, 26, 27, 28, 29, 30 con tutti i metodi tranne il 12 (figura 6.9). Questo era previsto già in fase di elaborazione in quanto il metodo 12 lancia un flag (fl2) ossia nessuno dei voxel selezionati dallo spettro e poi raggruppati supera la terza fase dei vincoli; il programma continua utilizzando per l'elaborazione della fase 2 i voxel non raggruppati. Per tale soggetto il metodo 11 è il migliore.

Il soggetto 16, affetto da lesione di tipo POCI a destra e PEM presente, dà una buona attivazione dell'area SII e dell'insula controlaterali allo stimolo (fette dalla 21 alla 24 e dalla 27 alla 33) rilevata da tutti e 4 i metodi, in particolare dal metodo 12 (figura 6.10).

Per quanto riguarda i soggetti sani, anche se si è visto dalla tabella 6.1 che l'indice AIC minore lo si ottiene con i metodi ad hrf fissa, dall'ispezione visiva dei risultati si nota che il soggetto 18 presenta una vasta attivazione sulle aree SM1 ipsilaterale e controlaterale al movimento e sull'area SMA con tutti i metodi; il metodo che in questo caso va peggio è il 10 in quanto rileva meno voxel (figura 6.15).

Il soggetto 19 presenta un'attivazione discreta delle aree SM1 controlaterale al movimento e SMA (dalla fetta 29 alla 36) con tutti i 4 metodi; i metodi migliori per questo soggetto sono 7 e 11.

Infine il soggetto 20, presenta con tutti i 4 metodi, attivazione delle aree insula e SII bilaterali (fette dalla 17 alla 20) e dall'area SI controlaterale allo stimolo (fette: dalla 24 alla 29) (figure 6.11, 6.12, 6.13 e 6.14). Il metodo migliore in questo caso è il 7, anche per gli altri le differenze sono minime.

È interessante osservare i differenti andamenti delle hrf specifiche per ogni soggetto ottenute dai metodi 11 e 12, tali curve sono riportate in figura 6.17, 6.18 e 6.19. Si nota che per i soggetti sani la latenza del picco positivo delle hrf varia dai 5 s ai 6 s,

mentre per i pazienti varia da 8 s a 10 s. Il metodo 12 tende a ritardare o anticipare l'hrf rispetto a quella del metodo 11.

Infine si possono confrontare tali curve con quelle fisse dei metodi 7, 8, 9, 10 (figura 6.16).

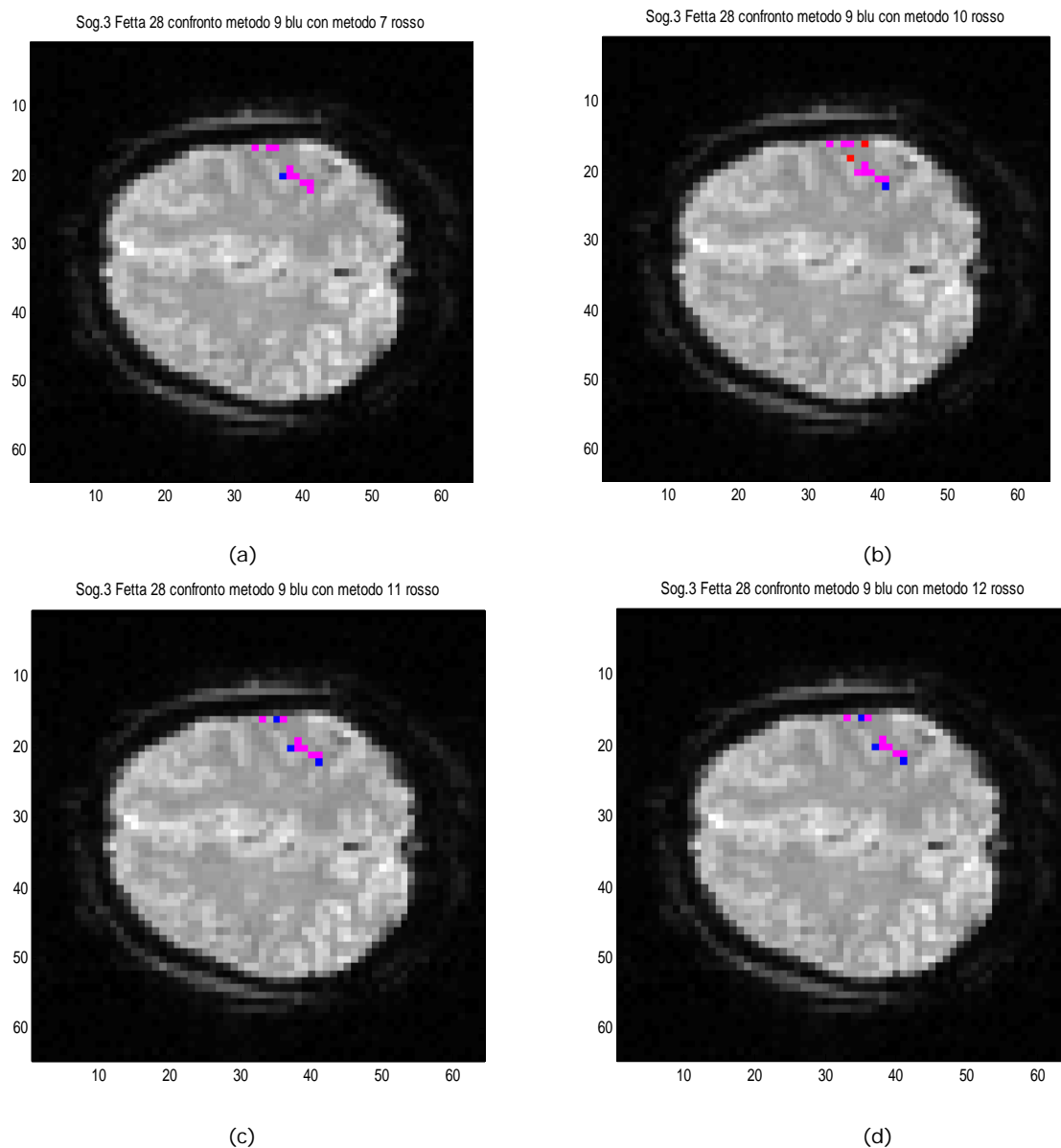


Fig.6.7: Soggetto 3, fetta 28 confronto tra: (a) metodo 9 e 7, (b) 9 e 10, (c) 9 e 11, (d) 9 e 12. Stessa colorazione delle figure precedenti.

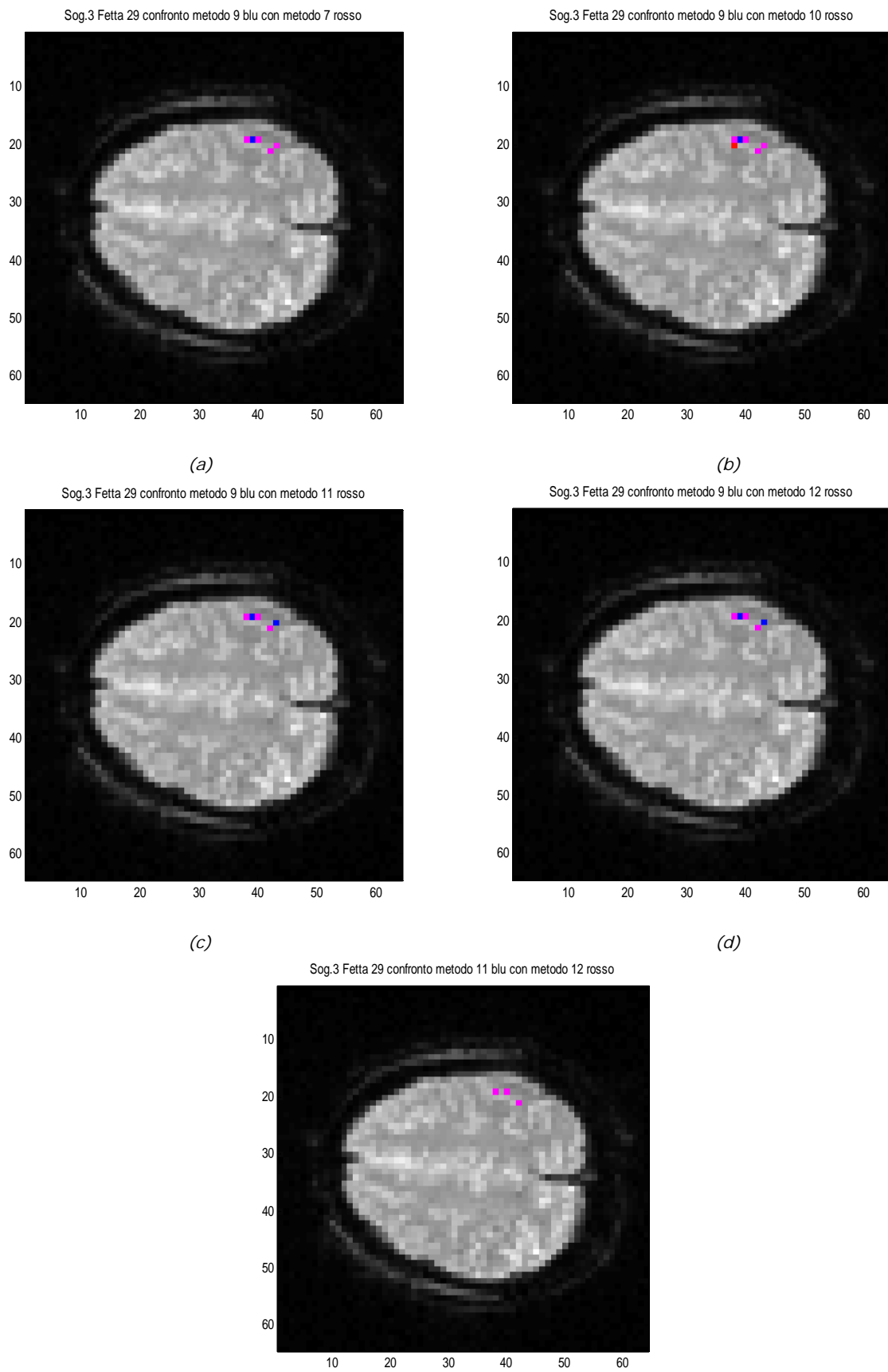


Fig.6.8: Soggetto 3, fetta 29 confronto tra: (a) metodo 9 e 7, (b) 9 e 10, (c) 9 e 11, (d) 9 e 12 ed (e) 11 e 12. Stessa colorazione delle figure precedenti.

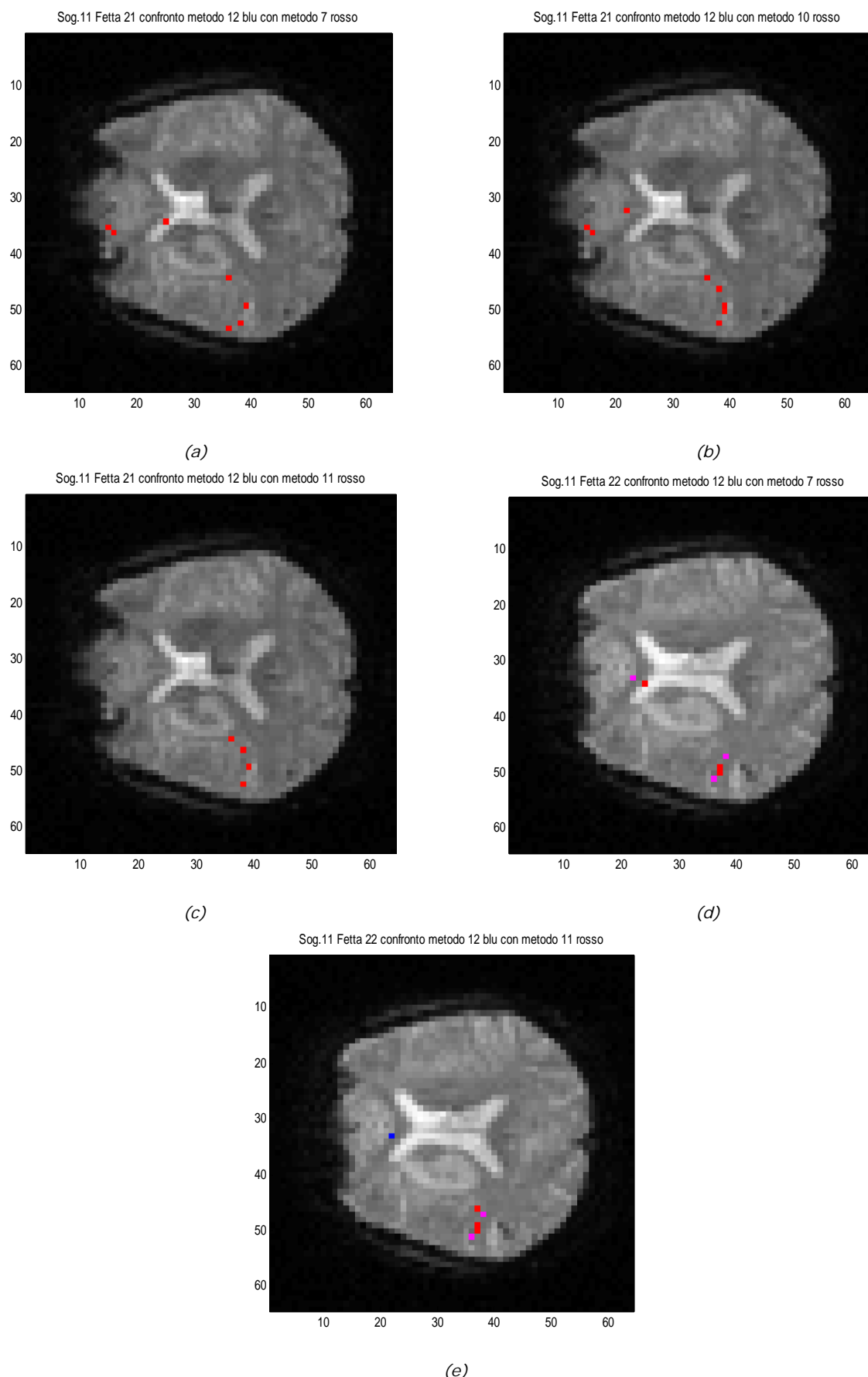
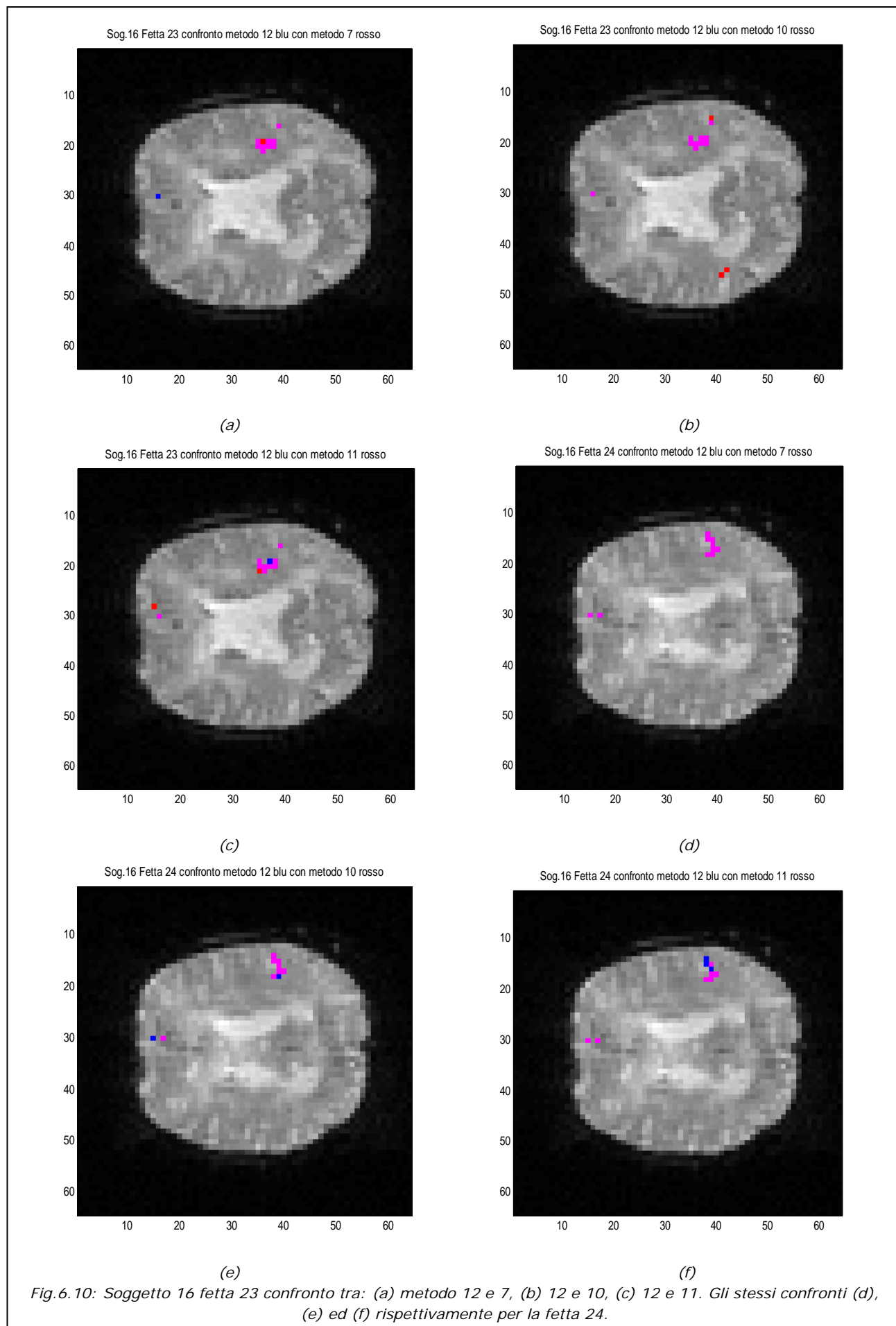


Fig.6.9: Soggetto 11 fetta 21 confronto tra: (a) metodo 12 e 7, (b) 12 e 10, (c) 12 e 11. Fetta 22 (d) metodo 12 e 7, (e) 12 e 11.



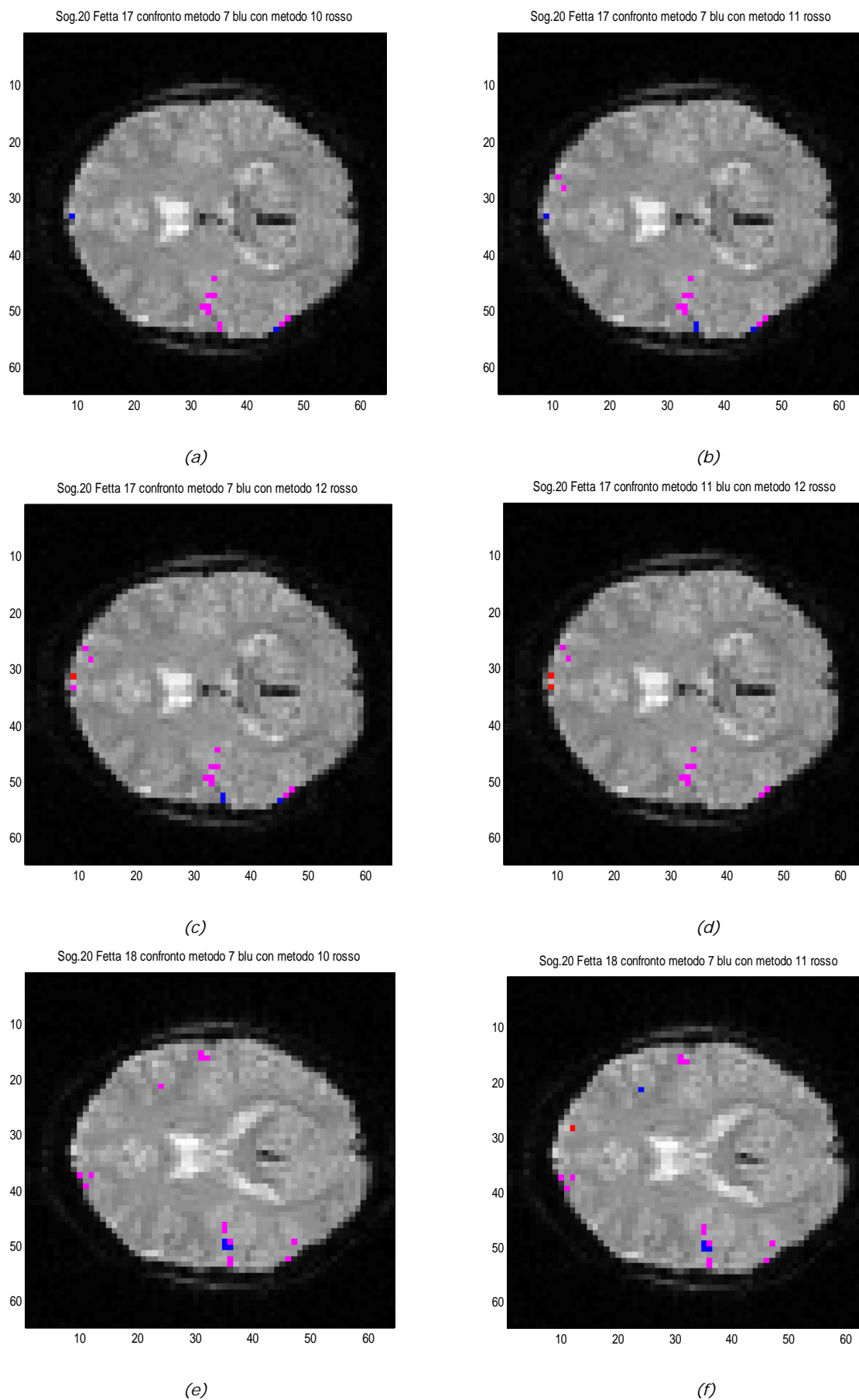


Fig.6.11: Soggetto 20 fetta 17 confronto tra: (a) metodo 7 e 10, (b) 7 e 11, (c) 7 e 12, (d) 11 e 12. (e) e (f) sono rispettivamente gli stessi confronti di (a) e (b), sulla fetta 18.

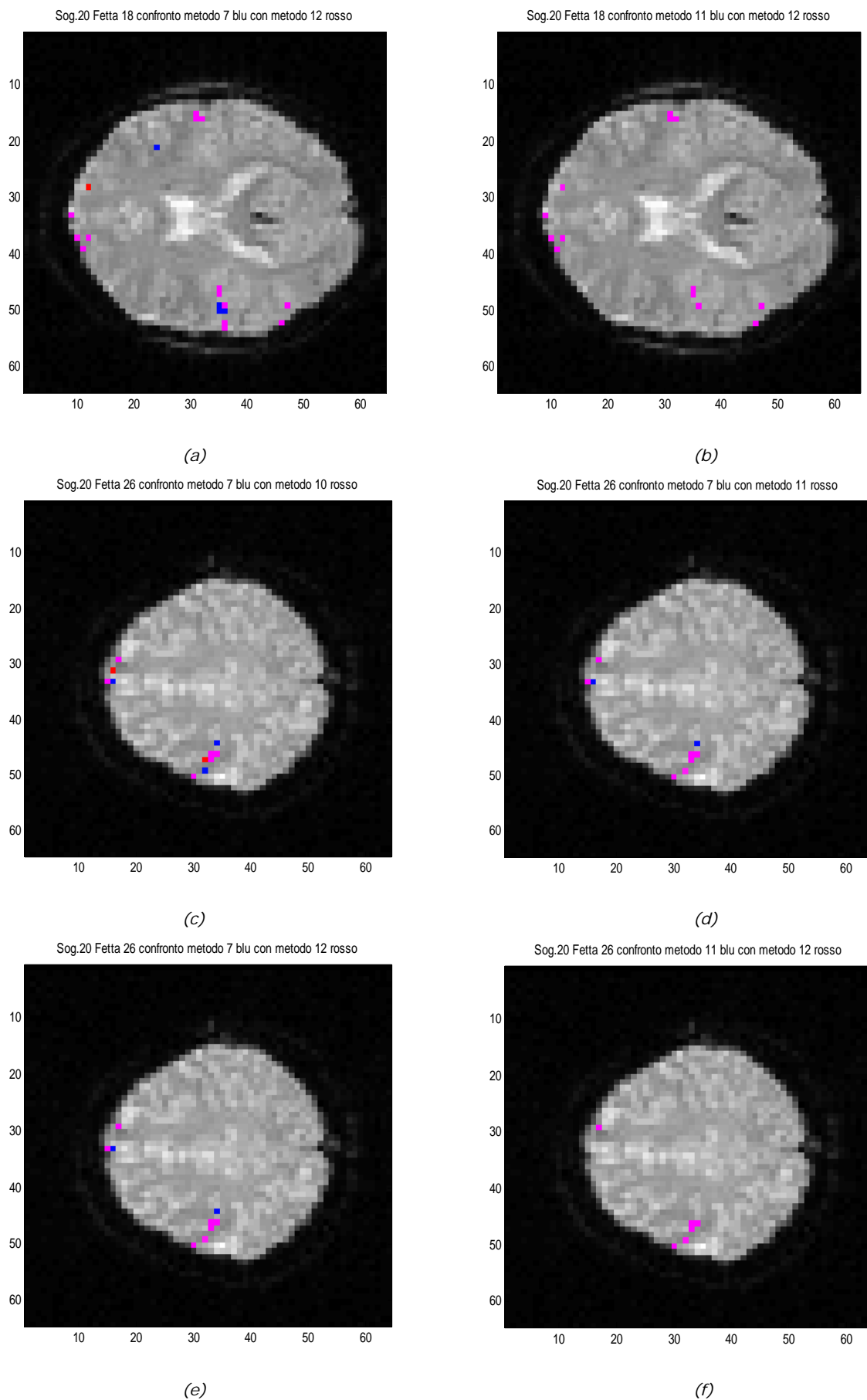


Fig.6.12: Soggetto 20 fetta 18 confronto tra: (a) metodo 7 e 12, (b) 11 e 12; fetta 26: (c) 7 e 10, (d) 7 e 11, (e) 7 e 12, (f) 11 e 12.

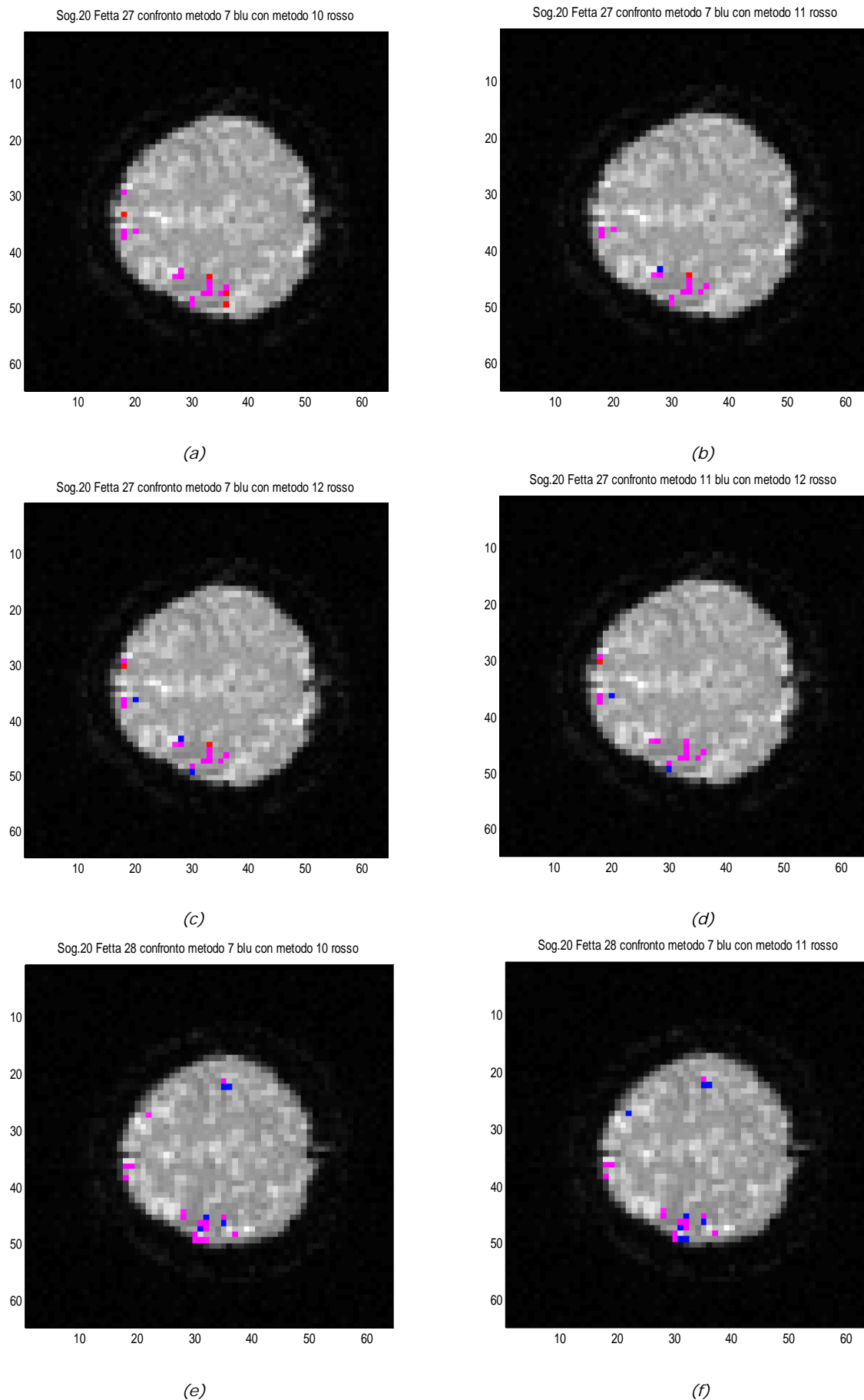
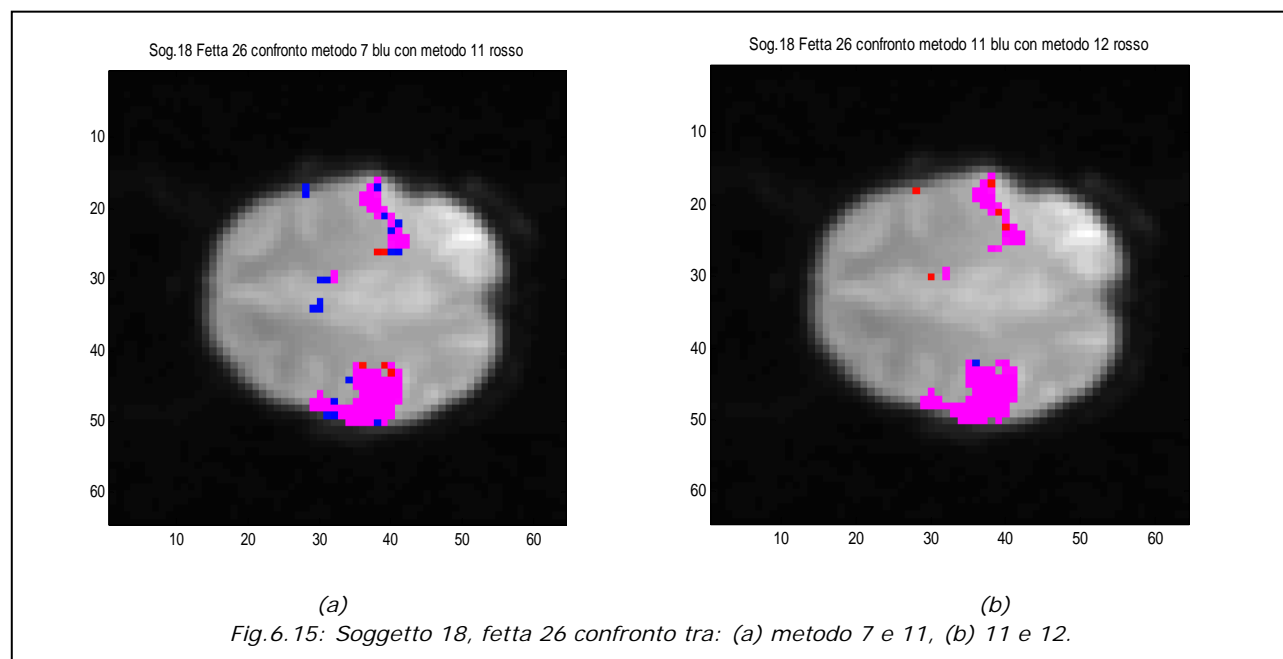
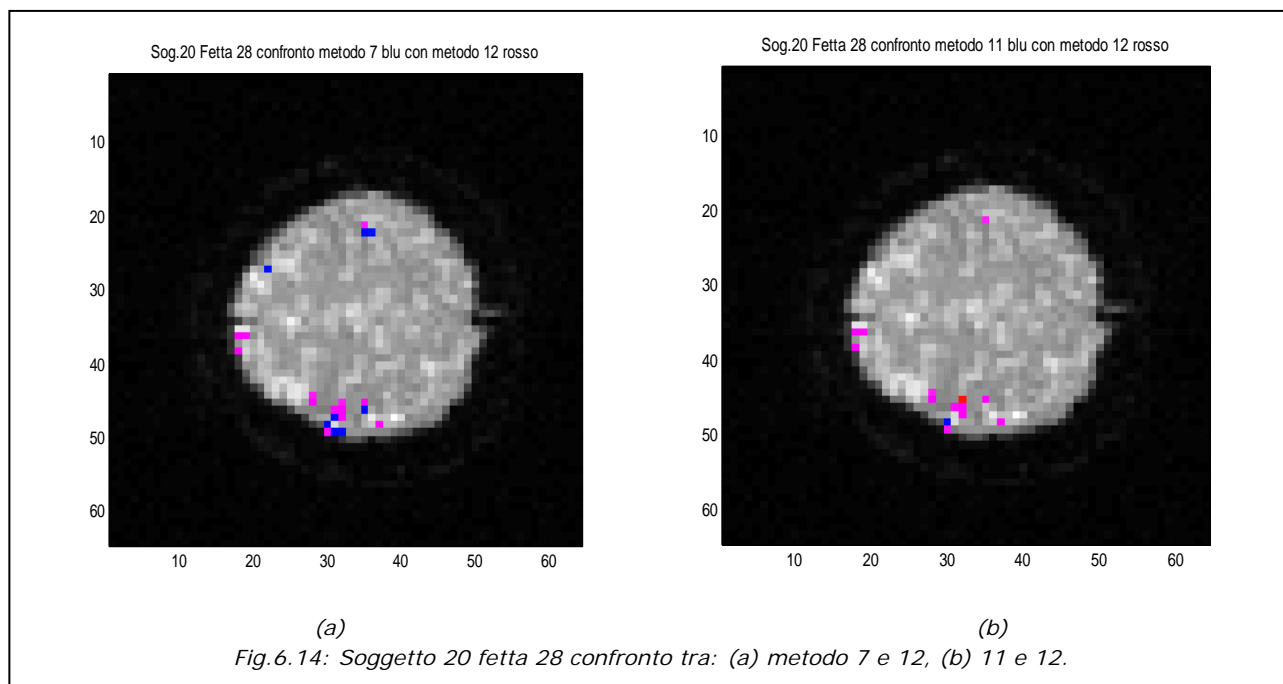
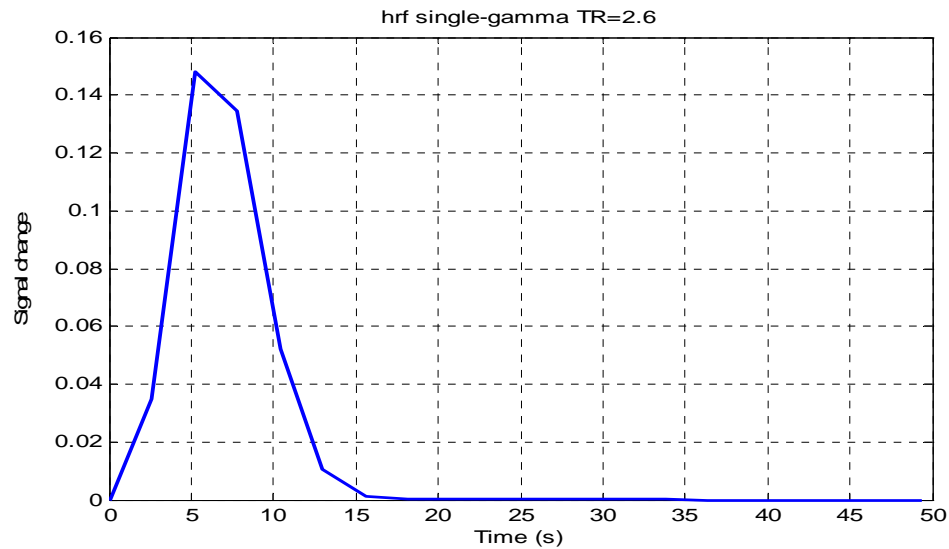
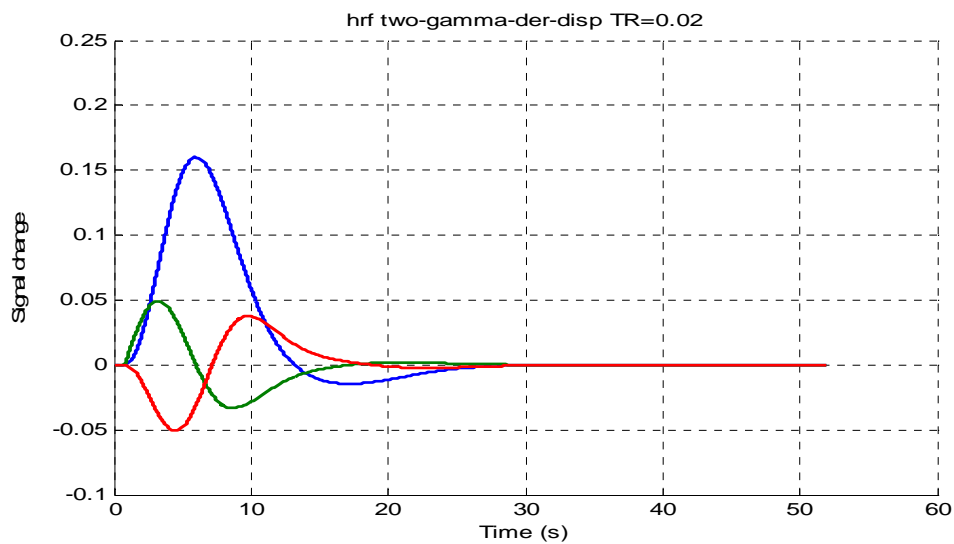


Fig.6.13: Soggetto 20 fetta 27 confronto tra: (a) metodo 7 e 10, (b) 7 e 11, (c) 7 e 12, (d) 11 e 12; fetta 28: (e) 7 e 10, (f) 7 e 11.

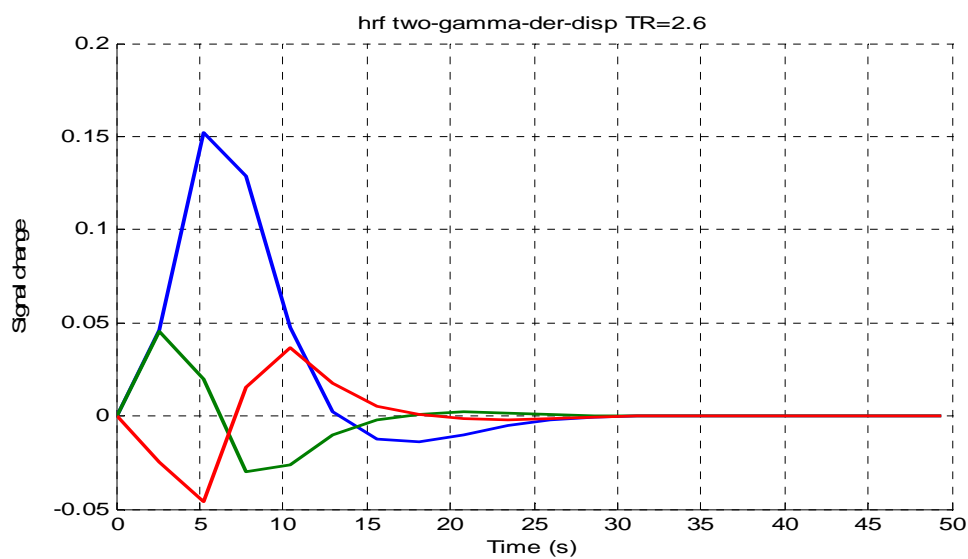




(a)



(b)



(c)

Fig. 6.16: (a) hrf single-gamma HW con passo di campionamento 2.6; (b) hrf two-gamma HW (blu), relativa derivata temporale (verde) e derivata dispersiva (rosso) con campionamento 0.02; (c) le stesse curve di (b) con campionamento 2.6.

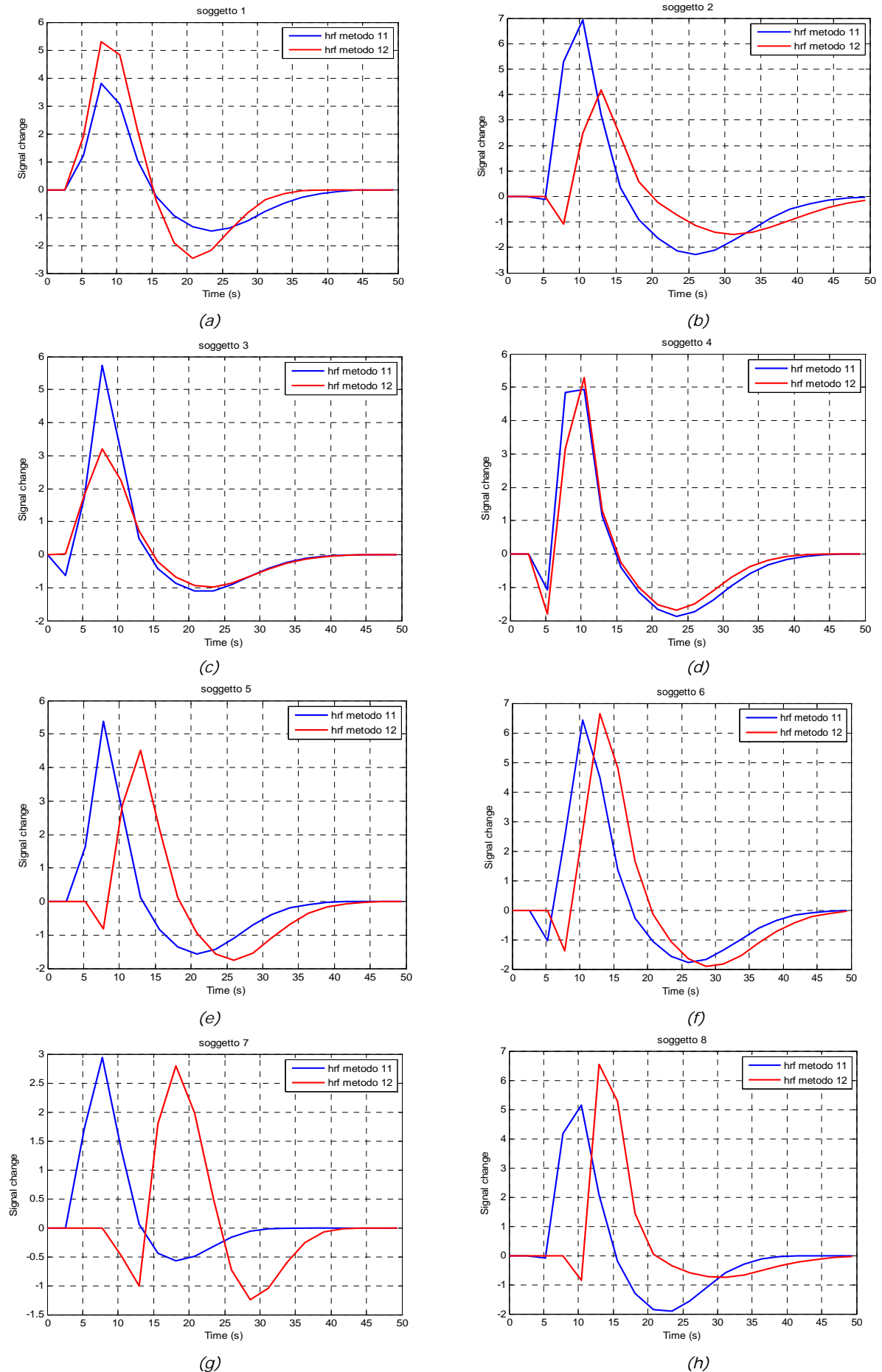


Fig.6.17: Confronto tra le hrf ottenute con il metodo 11 (blu) e quelle del 12 (rosso) per i soggetti 1-8 campionamento 2.6.

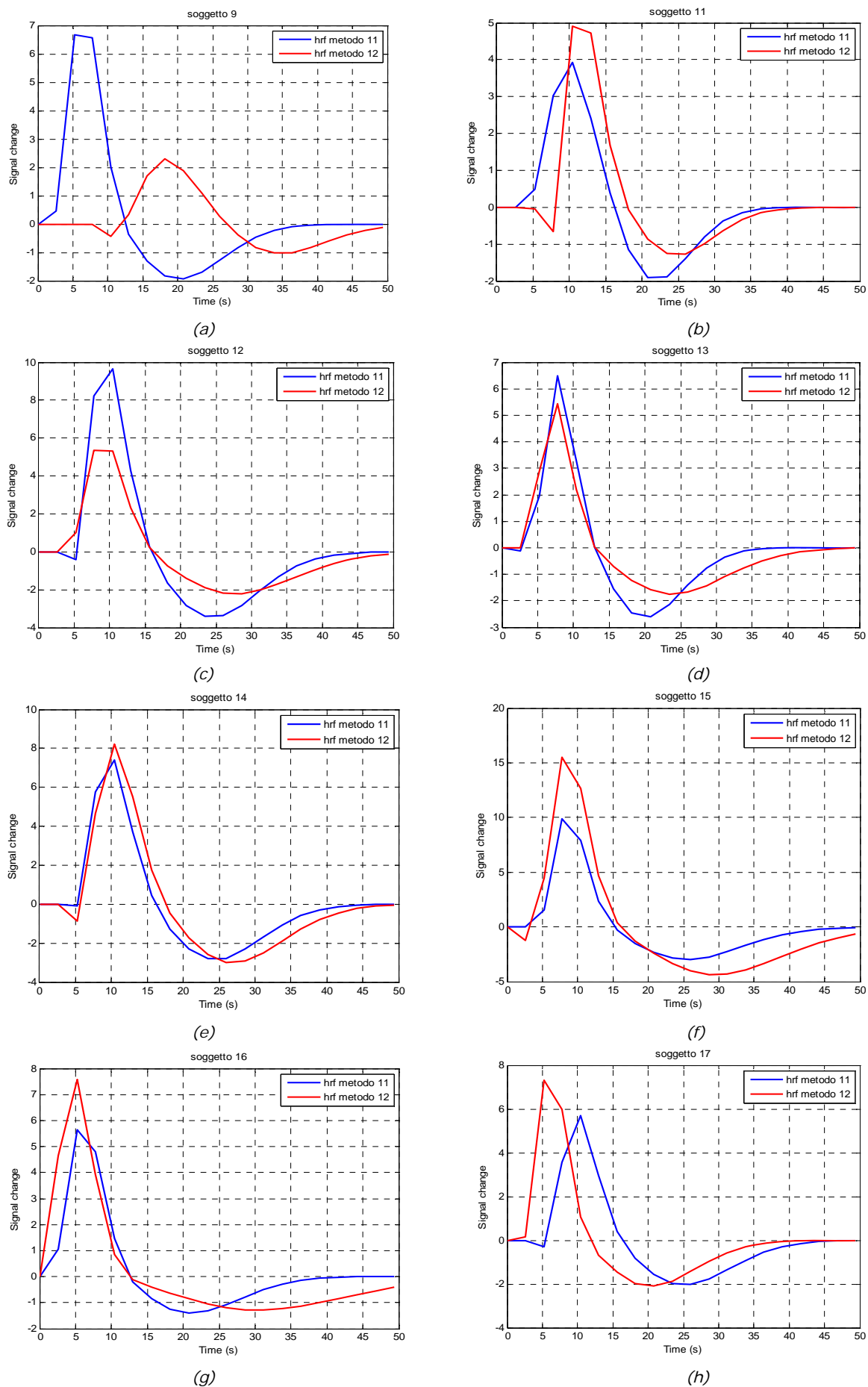
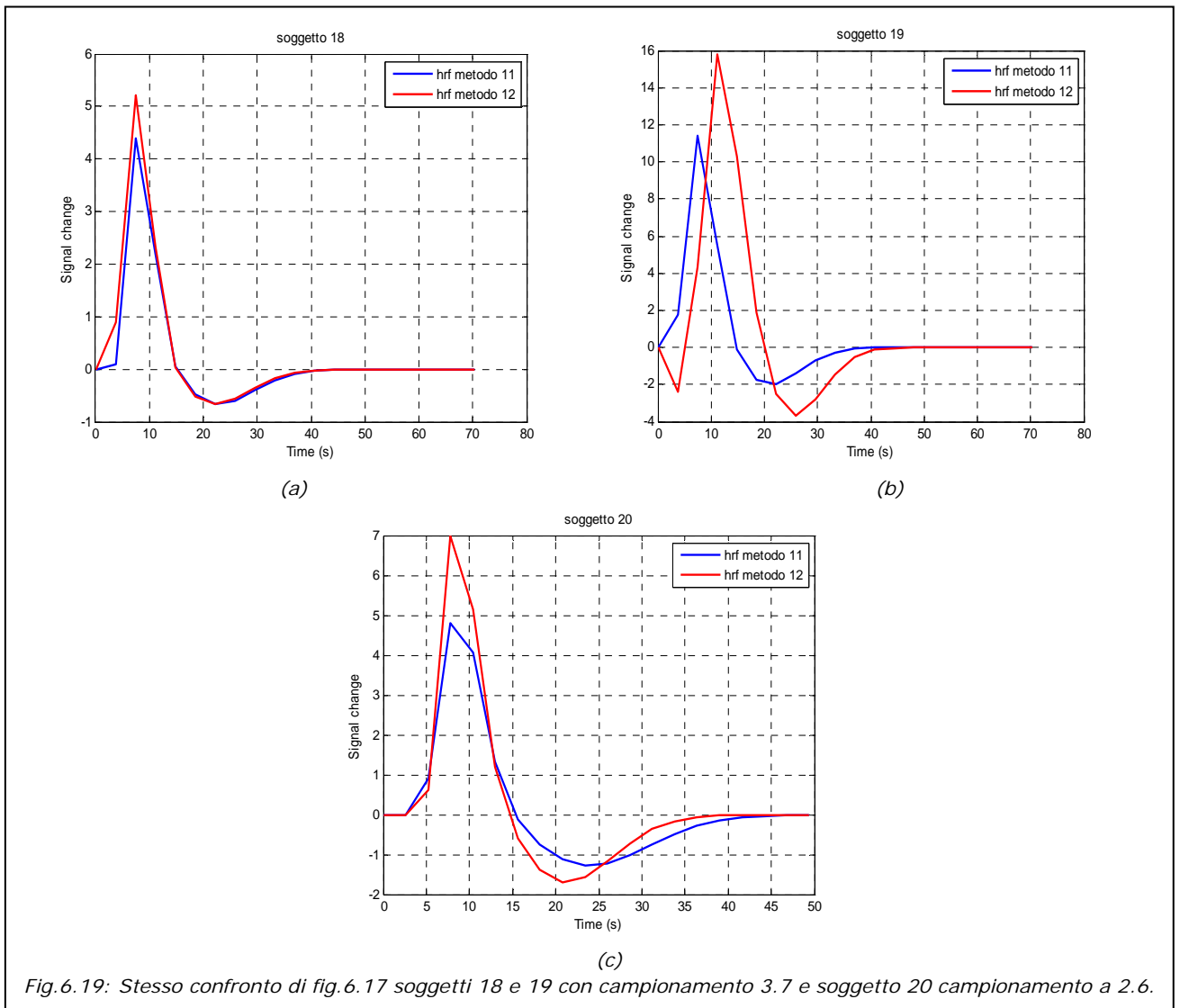


Fig. 6.18: Stesso confronto di fig. 6.17 soggetti 9-17.



Conclusioni

In tabella 7.1 è riportato un riassunto del comportamento dei vari metodi per ciascun soggetto. Dall'analisi risulta che i metodi migliori sono il metodo 11 e il 12 rispettivamente con 5 soggetti ciascuno, mentre i metodi 7 e 10 sono i migliori in 2 soggetti ciascuno. In ogni caso per tali soggetti si ha attivazione anche con i metodi 11 e 12.

Per la scelta del migliore tra il metodo 11 e 12, si deve tener presente che il 12 ha un'elaborazione notevolmente più veloce nella prima fase di selezione e raggruppamento; si consiglia quindi di processare i dati di un ipotetico nuovo paziente con tale metodo. Nel caso in cui tale metodo lanci il flag 2 (soggetto 11 del data set), risulta non attendibile quindi si consiglia di procedere con il metodo 11 più robusto.

Tab. 7.1: Tabella riassuntiva dell'attivazione nei quattro metodi migliori.

Soggetti	7)HRF B HW	10)HRF GFTD HW	11)HRF media dei parametri modificata	12)Spettro +lsqnonlin
1	attiva male	attiva	attiva	attiva
2	attiva	attiva	attiva	attivazione migliore
3	attiva	attivazione migliore	attiva	attiva
4	attivazione rumorosa	attivazione rumorosa	attivazione rumorosa	attivazione rumorosa
5	-----	-----	-----	-----
6	attivazione rumorosa	attivazione rumorosa	attivazione rumorosa	attivazione rumorosa
7	-----	-----	-----	-----
8	-----	-----	-----	-----
9	-----	-----	-----	-----
10	-----	-----	-----	-----
11	attiva	attiva	attivazione migliore	non attiva
12	-----	-----	-----	-----

13	non attiva	non attiva	attiva male	attiva male
14	attiva	attiva	attivazione migliore	attiva
15	attivazione rumorosa	attivazione rumorosa	attivazione rumorosa	attivazione rumorosa
16	attiva	attiva	attiva	attivazione migliore
17	attivazione rumorosa	attivazione rumorosa	attivazione rumorosa	attiva con rumore ridotto
18	attiva	attiva	attiva	attiva
19	attivazione migliore	attiva	attivazione migliore	attiva
20	attivazione migliore	attiva	attiva	attiva

Analizziamo ora i pregi e limiti dei due metodi proposti. Il metodo 12 è veloce e soprattutto per la stima dell'hrf specifica per il paziente utilizza una selezione indipendente da un'hrf prefissata. Il suo limite però è quello di non poter essere applicato a protocolli che non hanno un periodo fisso, come gli *event-related*, ma soltanto a protocolli *blocked*. Per quanto riguarda il metodo 11 può essere applicato a qualunque protocollo e pur essendo meno veloce del metodo 12 ha il pregio di essere computazionalmente meno oneroso dei metodi ad hrf fissa. Tali metodi infatti processano i dati tante volte quante sono le hrf fisse per rilevare quella che dà i risultati migliori. Il punto critico del metodo 11 è la selezione iniziale dei voxel su cui, attraverso la function *lsqnonlin*, si rileva l'hrf specifica. Tale selezione utilizza un'hrf fissa, che in parte condiziona la successiva stima dell'hrf specifica. Un aspetto rilevante è lo sviluppo di un metodo applicabile a protocolli *event-related* che determini una selezione iniziale indipendente da un'hrf base.

Bibliografia e sitografia

- [1] G. Valli, G. Coppini, Bioimmagini, Bologna: Pàtron, 2005, pp. 205-228.
- [2] A. Bertoldo, lucidi del corso di Neuroingegneria.
- [3] A. Ruggeri, lucidi del corso di Strumentazione Biomedica 2.
- [4] L. Calligaro, A. Mantovani, Fondamenti di chimica per ingegneria, Padova: Cortina, 2001, p. 7.
- [5] K. Friston et al., Statistical Parametric Mapping, Elsevier, 2007, pp. 101-125, 178-190.
- [6] C.H. Liao, K.J. Worsley, J.-B. Poline, J.A.D. Aston, G.H. Duncan, A.C. Evans, 'Estimating the Delay of the fMRI Response', NeuroImage vol 16, pp. 593-606, 2002.
- [7] D.A. Handwerker, J.M. Ollinger, M.D. Esposito, 'Variation of BOLD hemodynamic responses across subjects and brain regions and their effects on statistical analyses', NeuroImage vol 21, pp. 1639-1651, 2004.
- [8] L. Lamberti, L. Mereu, A. Nanni, Nuovo Matematica uno, Milano: Etas, 1998, pp. 588-594.
- [9] Anil K. Jain, Fundamentals of digital image processing, 1989, pp. 150-151.
- [10] P. Jezzard, P.M. Matthews, S.M. Smith, Functional MRI an introduction to methods, Great Britain, Oxford University Press, 2001, pp. 251-258.
- [11] G. Toffolo, lucidi del corso di Elaborazione dei segnali biologici.
- [12] M.P. Saccomani, lucidi del corso di Bioimmagini.
- [13] Help di Matlab versione 2007b.
- [14] http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm.
- [15] http://it.wikipedia.org/wiki/Risonanza_magnetica_funzionale.
- [16] www.spread.it/node/458.
- [17] S. Molinelli, 'Lo Stroke Epidemiologia, Patofisiologia, Classificazione', Febbraio 2007.
- [18] Codice Matlab per l'implementazione del GLM di S.F. Storti, 2007.
- [19] S.F. Storti, Metodi e modelli per la generazione di mappe di attivazione da immagini fMRI, Tesi di laurea, A.A. 2006/2007, cap. 3, 4, 5.

- [20] C.Cobelli, lucidi del corso di Modelli e controllo dei sistemi biologici 2.
- [21] F.E.Turkheimer, Rainer Hinz,.V.J.Cunningham, ‘On the undecidability among kinetic models:from model selection to model averaging’, Journal of Cerebral Blood Flow & Metabolism, vol. 23, pp.490-498,2003.
- [22] C.Cobelli, R.Bonadonna, Bioingegneria dei sistemi metabolici, Bologna: Pàtron, 1998, pp.241-245
- [23] A.C.Guyton, J.E.Hall, Fisiologia Medica, Napoli:EdiSES, 2005, pag 650.